Group Representation Learning for Group Recommendation

by

Sarina Sajadi Ghaemmaghami

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada

January 2021

Thesis Examination Information

Submitted by: **Sarina Sajadi Ghaemmaghami**

Master of Science in **Computer Science**

Thesis title: Group Representation Learning for Group Recommendation

An oral defense of this thesis took place on December 9, 2020 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Amirali Salehi-Abari
Research Supervisor	Dr. Amirali Salehi-Abari
Examining Committee Member	Dr. Faisal Qureshi
Examining Committee Member	Dr. Julie Thorpe
Thesis Examiner	Dr. Karthik Sankaranarayanan

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Group recommender systems facilitate group decision making for a set of individuals (e.g., a group of friends, a team, a corporation, etc.). Existing group recommendation methods mostly learn group members' individual preferences and then aggregate them into a group preference. This thesis takes a different approach. We focus on making recommendations for a new group of users whose preferences are unknown, but we are given the decisions/choices of other groups. By formulating this problem as *group recommendation from group implicit feedback*, we focus on two of its practical instances: Given a set of groups and their observed decisions, *group decision prediction* intends to predict the decision of a new group of users whereas *reverse social choice* aims to infer the preferences of those users involved in observed group decisions. These two problems are of interest to not only group recommendation, but also to personal privacy when the users intend to conceal their personal preferences, but have participated in group decisions. To tackle these two problems, we propose and study DeepGroup—a deep learning approach for group recommendation with group implicit data. We empirically assess the predictive power of DeepGroup on various real-world datasets, group conditions (e.g., homophily or heterophily), and group decision (or voting) rules. Our extensive experiments not only demonstrate the efficacy of DeepGroup but also shed light on the privacy-leakage concerns of some decision making processes.

Keywords: Group Recommendation; Social Choice; Deep Learning; Group Implicit Feedback; and Representation Learning;

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Sarina Sajadi Ghaemmaghami

Statement of Contributions

I hereby certify that I have been the primary contributor of this thesis by developing the algorithms, implementing them, and designing the experiments. I have also written most content of this thesis. However, some texts of this thesis are borrowed from the conference paper jointly coauthored by my thesis supervisor Dr. Amirali Salehi-Abari and me.

Acknowledgements

First of all, I would like to express my gratitude and appreciation to my supervisor, Dr. Amirali Salehi-Abari for providing his guidance, support, encouragement, and insights without whom I would not have been able to complete this research. I would like to say a special thanks to my parents, Majid and Mojgan, my brother Parsa, and my boyfriend Roozbeh for the encouragement, assurance, and unconditional love during this journey. Finally, I would like to thank Ontario Tech University, the entire faculty, and my supportive friends.

Contents

List of Figures

List of Symbols

d, k, l, m, n .... Variables.

$\mathcal{U}$ .... Set of users.

$\mathcal{A}$ .... Set of alternatives (items).

$\mathcal{G}$ .... Set of groups of users.

E .... Set of all user embeddings.

$\boldsymbol{Y}$ .... Group-item interaction matrix.

f .... Likelihood function.

$\boldsymbol{\theta}$ .... Model parameters.

g .... Vector representation of groups.

$\mathbf{U}$ .... Latent representation matrix of users.

$\mathbf{V}$ .... Latent representation matrix of items.

$\mathbf{q}$ .... Group representation vector.

List of Abbreviations

AE .... Autoencoder.

AGREE .... Attentive Group Recommendation.

AM .... Attentional Models.

BGEM .... Bipartite Graph Embedding Model.

BPR .... Bayesian Personalized Ranking.

CDL .... Collaborative Deep Learning.

CF .... Collaborative Filtering.

CKE .... Collaborative Knowledge-base Embedding.

CNN .... Convolutional Neural Network.

CVAE .... Collaborative Variational Autoencoder.

DL .... Deep Learning.

DNN .... Deep Neural Network.

eALS .... element-wise Alternating Last Squares.

EMF .... Explicit Matrix Factorization.

JoVA .... Joint Variational Autoencoder.

MF .... Matrix Factorization.

MLP .... Multilayer Perceptron.

MAGRM .... Multi Attention-based Group Recommendation Model.

ML .... Machine Learning.

MMM .... Combined Aggregator (Mean, Max, Min).

RNN .... Recurrent Neural Network.

SDAE .... Stacked Demonising Auto Encoder.

SIGR .... Social Influence-based Group Representation.

VAE .... Variational Autoencoder.

Chapter 1

Introduction

Nowadays, by immense growth of group social activities in our society, addressing group decision problems has become crucial. Group decision problems are prevalent, ranging from high-stake decisions (e.g., elections, the board of directors' decisions, etc.) to casual decisions (e.g., deciding on a restaurant, movie, or vacation package for a group of friends). In recent years, making a group decision in both online social platforms and face-to-face settings has become challenging due to the overwhelming availability of options, and the complex and conflicting preferences of group members. In such scenarios, group recommender systems play an integral role in facilitating group decision making by recommending a set of items (or a single item) to a group of people such that the recommendation is satisfactory for all the members. The applications of group recommender systems are diverse, such as tourism [1], music [2], crowdfunding [3], news/web pages [4], TV programs [5], and movies [6].

Group recommendation methods (e.g., [7] [8] [9] [10] [11] [12] [13]) mainly assume that (1) user preferences can be elicited (or inferred) and then aggregated into group preferences or (2) group preferences are partially observed/elicited. Many of these systems usually encompass two key integrated components: *preference assessment* and *preference aggregation*. The *preference assessment* component focuses on understanding group members' preferences. Two common approaches for preference assessment are *preference elicitation* (e.g., [14]–[16]) via asking relevant queries for revealing user preferences and *preference learning* from historical data represented either in the form of rankings (e.g.,[11], [17], [18]) or user-item interaction data (e.g.,[7]–[10], [19]). The *preference aggregation* component aggregates individuals' inferred (or elicited) preferences into group preferences (or decisions). These aggregation methods are usually well-studied social choice functions (or group consensus functions) [20], [21] or learned using attention mechanisms of deep learning (e.g.,[13], [22], [23]).

Group recommender systems that utilize deep neural networks for the process of recommendation to a group of users have become popular in recent years. The main contribution of existing group recommender systems that use deep neural networks is that they employ deep learning for learning group representations and predicting group recommendations. These models mainly use individual interactions of group members for the learning process. In other words, they focus on learning users' representations from user-item interactions (individual preferences of users) and then learn groups' representations by aggregating users' representations. The group recommendation also would be learned based on its members' preferences. However, the main drawback of these models is their poor performance in dealing with incomplete input data. For instance, in real-world group decision-making problems, individual preferences of group members may be inaccessible due to privacy, and only the groups' previous decisions are available. Moreover, in some scenarios, specific features of groups/users including auxiliary information that improve learning efficiency are not accessible. To address these issues, a group recommender system model is required to learn the group representations and predict group decisions when individual preferences of group members are unobserved and only some other groups' decisions are available.

1.1 Contributions

This thesis initiates a study of group recommendation for a *cold start* group whose members' personal preferences are not explicitly accessible via preference elicitation or user-item interaction data. But, we are given certain *implicit feedback* on some other groups in the form of their membership and decisions. Hence, we focus on a new problem: *group recommendation from group implicit feedback*. The applications of this problem are prevalent in our daily lives. For instance, consider recommendations of restaurants, vacation packages, activities to a group, when we have observed restaurants, places, activities in which some group members have visited/participated with their family members or friends. One can note two special instances of group recommendation with group implicit feedback: Given a set of groups and their observed decisions, *group decision prediction* intends to predict the decision of a new group of users whereas *reverse social choice* aims to infer the preferences of a user involved in observed group decisions. The latter special case is derived when the new group is a singleton set. In addition to group recommendation, these two special cases are of high importance for assessing privacy leakage. Imagine those users who intend to conceal their personal preferences on a sensitive issue (e.g., promotion, social injustice issues, etc.), but have participated in group decisions on these topics with publicly known decisions.

To address the mentioned group recommendation problem, we propose *DeepGroup*—a deep neural network for learning group representations and decision making. The contributions of this thesis include:

1. **DeepGroup**. We propose DeepGroup, a deep neural network (DNN) model for learning group representations and predicting group decisions. By utilizing a multilayer perceptron (MLP) network, our proposed model can efficiently learn (new) group-item interactions, just by observing the decision history of previous groups. DeepGroup not only can be employed for solving our group recommendation problem but also can be easily extended for other groups/personal recommendation tasks.
2. **Learning Group Representations**. We present a novel method for learning user representations given group-item interactions as input data. This approach differs from existing methods that learn user embedding from user-item interactions. We then propose different methods for integrating group representations from users' representations.
3. **Experiments**. We conducted extensive experiments to evaluate the effectiveness of DeepGroup for both group decision prediction and reverse social choice. Our findings confirm the superiority of DeepGroup over a set of benchmarks for both problems over various datasets. In our experiments, we also study how different group decision rules (or group decision-making processes) might affect the performance of DeepGroup. Our findings show that DeepGroup excels (compared to benchmarks) regardless of how group decisions are made. In the reverse social choice task, the prominence performance of DeepGroup varies for different voting rules. This is an interesting observation regarding privacy. Experiment results

indicated that despite requiring the least personal preference data (i.e., only top choice) for decision making, a first-past-the-post voting rule can have the highest privacy leakage.

1.2 Thesis Organization

This thesis is organized as follow:

Chapter 2 describes some concepts of group recommendation with implicit feedback, required deep learning methods in our research, and the social choice theory. In **Chapter 3**, we review the existing work in the domains of group recommender systems, social choice and preference learning, and recommender systems and neural networks. In **Chapter 4**, we detail our proposed group recommendation problem, our approach for tackling the problem, and varying components involved in the structure of DeepGroup. In **Chapter 5**, we describe our variant experiments for evaluating our proposed model, the benchmarks, and the comparison of prediction accuracy of DeepGroup and benchmarks. Finally, we conclude our work and present future directions in **Chapter 6**.

Chapter 2

Background

This chapter reviews some foundational concepts used in this thesis. We first describe group recommendation with implicit feedback, then review the practical deep learning concepts utilized in our model, and finally describe some basic notions of social choice theory. In this thesis, we focus on learning groups' representations and making group recommendation from implicit feedback data by utilizing deep neural networks. We assume consensus decisions of the groups provided in the implicit data is derived from applying different social decision rules. Therefore, reviewing the background of these concepts could be advantageous for understanding our contribution.

2.1 Implicit Feedback Data

There are two important types of input data that recommender systems mostly rely on: *explicit* and *implicit* feedback data. Recommender systems can infer user/group preferences from explicit feedback data (e.g. ratings, comments, etc.) of preferences. Implicit feedback refers to clicks, watching data, queries, purchase history and other indirect activities of the users that could reveal his preference in the system. More precisely, implicit feedback data refers to the information which is not provided intentionally and is gathered from available data streams. However, explicit feedback data relies on the information that is provided intentionally, for example through surveys and membership registration forms.

Rating systems are the important representative of explicit feedback data. The most popular methods in this system are 5-star rating and binary thumbs up/down. For instance, Amazon is collecting users' preferences by capturing their (5-star) rating on the items, while Netflix and YouTube rely on binary like/dislike. The well-known Netflix Prize competition provided a large dataset of users and their ratings on movies (previously Netflix employed a 5-star rating system). Some powerful recommender systems utilized the Netflix explicit feedback dataset for recommendation [24] [25] [26].

Due to the lack of high-quality explicit feedback data, the challenges in collecting this type of data, and data sparsity of explicit feedback, implicit feedback has attracted more attention in recent years. Employing implicit feedback data for collaborative filtering (CF) methods appeared by presenting a TV recommender system in which the input data (that is considered as implicit feedback) is the number of times that each user has watched a TV program completely [27].

Two challenges in dealing with implicit feedback data are the sparsity of negative samples and their weak performance in online platforms compared to the offline settings. He et al. [28] proposed a new algorithm based on the element-wise Alternating Last Squares (eALS) technique for addressing these two issues in learning matrix factorization (MF) models from implicit feedback data. To address the first issue, they consider an item popularity-aware weighting system over the missing data (negative example feedback). This means that the negative samples do not have the same weight in the system. To address the latter issue, they propose a novel learning method for learning online real-time data.

2.2 Deep Learning and Neural Networks

Deep learning (DL) is a subset of machine learning (ML). Deep neural networks are specific architectures of deep learning that use neurons for transmitting data from input to output layers. Based on the type of the problem, deep neural network models with different structures would be trained to approximate any continuous function to any desired precision. In recent years, deep neural networks have been widely used in variety of tasks such as image recognition [29], speech recognition [30], natural language processing [31], and even in recommendation as a collaborative filtering method [32], session-based recommendation [33], etc. In recent years, deep neural networks in designing recommender systems have shown promising results.

2.2.1 Deep Learning Approaches in Recommender Systems

One of the successful approaches in recommender systems is collaborative filtering. In traditional collaborative filtering (CF) based models, ratings of items given by users are mainly used for learning the model. The main drawback of this model is the sparsity of the rating matrix (a matrix that captures the ratings of items given by users) which leads to low performance of the model. This sparsity could happen in some scenarios when the items' rating history is not complete or when we have new users which refer to the cold start problem. To address this issue, several deep learning methods are proposed that could enhance the performance of item latent representation.

The practical perspective of deep learning in recommender systems is learning deep representations from input data. There exist different architectural paradigms of deep neural networks that have been widely used in recommender systems so far. Some of the most popular neural networks (supervised and unsupervised) employed in recommender systems are:

- Multilayer Perceptron (MLP)
- Autoencoder (AE)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Attentional Models (AM)

The most important reasons that make employing deep neural networks in recommender systems significant could be integrated as: the nonlinear transformation behavior, effective representation learning, supporting sequential modeling tasks, and flexibility in the implementation of neural networks [34].

2.2.2 Multilayer Perceptron (MLP)

Multilayer Perceptrons (MLPs) or deep feedforward networks are one of the basic and significant deep learning models. These models would learn model parameters through multiple hidden layers (the output of training data is hidden in these layers) to approximate a target function. The chain structure Multilayer Perceptrons (MLPs) can be formulated as follow:

$$\begin{aligned}
\mathbf{h}^{(1)} &= g^{(1)}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) \\
\mathbf{h}^{(i)} &= g^{(i)}\left(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}\right) \qquad\qquad \text{for i} > 1,
\end{aligned}$$

Where $\mathbf{x}$, $\mathbf{h}^{(i)}$, $\mathbf{W}^{(i)}$, $\mathbf{b}^{(i)}$, and $g^{(i)}(.)$ are the input, the hidden unit vector, the linear transformation weight matrix, bias vector, and non-linear activation function for layer i, respectively. The output layer of this network can be defined as:

$$\hat{y} = \phi\left(\mathbf{h}^{(o)}\right),$$

Where $\hat{y}$, ϕ, and $\mathbf{h}^{(o)}$ are the output, activation function of the output layer, and the output vector of the last hidden layer respectively.

Two popular examples of non-linear activation functions that we use in this thesis are $Relu(x) = \max(0, x)$ and Sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$. During learning the network, the gradient of the cost function will be minimized through a back-propagation algorithm. The cost (loss) function is a function that specifies the error of predicted values for model inputs. Based on the type of the problem, and the target function, different cost functions may be selected for the model. Figure 2.1 shows the architecture of a basic Multilayer Perceptron (MLP) network.

As we deal with a supervised learning task, we adopted a Multilayer Perceptron (MLP) network which is a feedforward neural network, containing one or more hidden layers. This network has the ability to learn the hierarchical representations. Applying non-linear activation functions provides the capability of capturing the non-

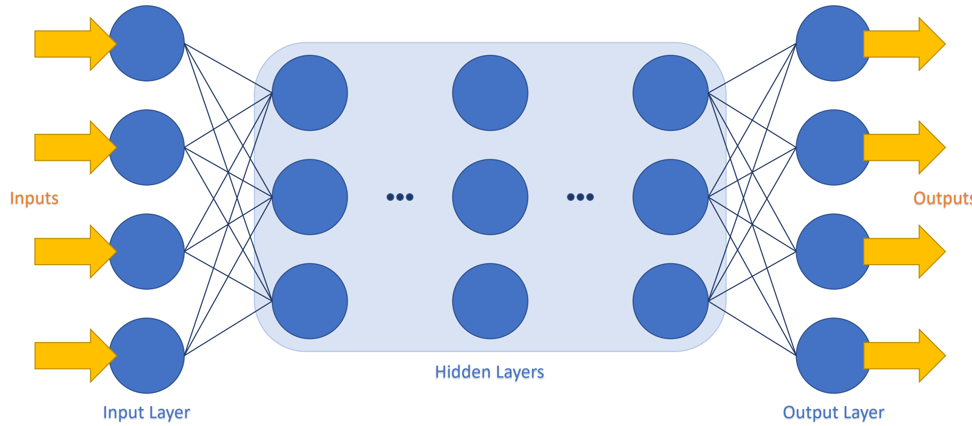


Figure 2.1: The architecture of Multilayer Perceptron (MLP), a fully connected feed-forward neural network consisting of the input layer, set of hidden layers, and the output layer. Based on type of the model, each layer could have different activation function and number of hidden units.

linearity features of the data. In our model, it is beneficial in extracting complex features of group-item interaction from the input data. Therefore, for our proposed group recommendation task, we utilize a Multilayer Perceptron (MLP) in our model.

2.3 Social Choice Theory

Social choice theory is a theoretical framework that aims to aggregate individual preferences into a consensus preference. In one general type of group recommendation process, the personal preferences of group members are integrated into a consensus decision. In these systems, one of the main steps is applying a social choice function to aggregate the individual preferences to generate preferences of the groups.

2.3.1 Voting Rules

The concept of preference aggregation refers to the theory of voting. This theory represents the process of selecting a candidate/alternative among several, which are preferred by individuals/voters in a finite group of people. The voting rule is expected to treat every voter or candidate equally to reach a collective decision.

One popular class of voting rule is scoring based rules. A scoring rule assigns a score vector to each user in the group. A score vector is in the form of $\mathbf{w} = (w_1, w_2, \ldots, w_m)$ that considers a weight for each alternative. Here, w_i is a real number scoring weight given to item i ranked i-th in the user's preference list. After summing the assigned weight scores of each alternative given by all the voters, the winner candidate (consensus choice) is the one with the highest score. Existing scoring rules and their corresponding score vectors include:

- Borda, $\mathbf{w} = (m-1, m-2, \ldots, 1, 0)$ (m is the number of candidates)
- Plurality, $\mathbf{w} = (1, 0, 0, \ldots, 0)$
- Anti-plurality, $\mathbf{w} = (1, 1, 1, \ldots, 0)$
- k-Approval, $\mathbf{w} = (1, 1, \ldots, 1, 0, 0, \ldots, 0)$ (k candidates with score of 1)
- Formula One Championship, $\mathbf{w} = (25, 18, 15, 12, 10, 8, 6, 4, 2, 1, 0, 0, \ldots, 0)$

The most commonly used social choice functions in voting systems are Borda and Plurality. The Borda decision rule (usually known as Borda scoring rule) will assign a weighted score $w(r_{ij}) = (m + 1 - r_{ij})$ to each ranked item. Here, r_{ij} is the ranking of item j assigned by user i, and m is the number of items. In the plurality rule, the winner is the item that has been the first ranked item of group members most times. In other words, the plurality weighted score is defined as $w(r_{ij}) = \mathbb{1}[r_{ij} = 1]$ where $\mathbb{1}[.]$ is an indicator function. This means that if the ranking of item j assigned by user i is equal to 1, the score of item j assigned by user i is equal to 1 and 0 otherwise.

2.3.2 Aggregation Strategies

Aggregation strategy refers to adopting a proper scheme to treat the group as a whole. In addition to scoring rules described in section 2.3.1, there are a lot of strategies that have been used for aggregating individual preferences in a voting procedure. For example, considering all the individuals in a group give a rating to all the candidates in the system, the *least misery* strategy, for each item assigns the minimum rating (given by the group members) to the group. This technique is advantageous for preventing the misery of group members. Mean of individual scores or *average strategy* is another commonly used strategies to achieve a consensus choice among group members. This method gets the average of ratings for each item (given by group members) and assigns it as the group rating. Another strategy is *most pleasure* which unlike least misery, considers the maximum rating among group members that leads to the maximum satisfaction of that user. Each of these strategies is adopted in different work based on the type of the proposed problem [35].

Chapter 3

Related Work

In this chapter, we review the related work on group recommender systems, computational social choice, preference learning, and deep learning recommender systems. Since our main contribution is learning group representations and the relationship between groups and items for making group recommendation, exploring the existing work in these areas is crucial before diving into our proposed approach.

3.1 Group Recommender Systems

Group recommendation is an emerging area in recommender systems [21], [36]. Finding relevant content of interest for socially connected individuals has been a progressive trend in recent years. Technological advances, the prevalence of team-build activities, and the advent of social networking websites and applications (e.g. Facebook, Instagram, etc.) have encouraged people to team bonding activities. Hence, group recommender systems have attracted more attention compared to personalized recommender systems in recent years. The group recommendation methods broadly fall into three categories: *Artificial profile*, *Profile-merging*, and *Recommendation/preference aggregation*. We review each of these categories in the following sections.

3.1.1 Artificial Profile

Artificial profile methods create a joint virtual user profile for a group of users to keep track of their joint revealed/elicited preferences.

In the profiling approach, the individual user profiles that indicate the interest level of each user for each item is manipulated by characterizing the significant information that captures the interest of the user [37]. In the next step, by considering the consumption behavior of each user (the behavior of the user in using products/items), the reference/target group profile is aggregated accordingly.

McCarthy [38] proposed MusicFX a recommender system that recommends music to groups of people in a fitness center. The key idea is to consider a group preference agent to capture the variant preferences of the group members in that shared environment. In this system, for each period of time, the virtual group profile is generated by computing the overall preferences of the present people in that period. This idea could be extended to any shared environment in which people gather together.

Based on the roles of users and their activity history, a subset of the group could participate in constructing individual virtual profiles. For example, a group recommender system called PolyLens [6], employs the least misery approach to make item recommendation for small groups of users (certain members of the larger group). For each group, the least misery strategy selects an item minimizing the lowest level of interest among group members.

Another strategy for building artificial group profiles is the majority-based approach. The most popular item among the group members is assigned to the preferences of individual virtual profiles in this strategy [39]. One significant drawback of these methods is ignoring social ties among the users in the groups. Studying the behavior of users in social networks and the possible impact of group members' preferences on the choice of other members demands more complex strategies and could result in a more accurate and satisfactory recommendation for groups.

3.1.2 Profile Merging

Profile-merging methods form a group profile by merging its members' user-item interactions, then the recommendation will be made based on a group profile [5].

Profile merging methods are mainly inspired by extending collaborative filtering (CF) methods that work efficiently in personalized recommendation tasks, to a group-based collaborative filtering approach. These methods appeared efficiently in real-world scenarios dealing with groups of people such as tourism recommendation systems, movie/restaurant recommender systems, TV program recommender systems, etc. For instance, in [40] a model for making TV program recommendation to a group of people is represented. Berkovsky and Freyne [40] compared different strategies for program recommendation to a group of viewers, and provided evidence that this strategy (merging individual profiles and integrating a group profile) indicates more efficient performance in addressing this type of problem compared to the existing methods.

Dwivedi and Bharadwaj also proposed a profile merging approach for recommending online resources to a group of learners [41]. They formed a unified group learner profile by taking advantage of learning styles, knowledge levels, and ratings of learners in a group. This merged group profile is representative of the preferences of all the learners. In the next step, a collaborative filtering (CF) approach is applied to this unified group profile for making recommendations.

3.1.3 Recommendation/Preference Aggregation

Recommendation/preference aggregation methods aggregate the recommendations (or inferred preferences) of group members into a group recommendation by using various *social choice functions* (also referred to as *group consensus functions*).

One of the usages of these methods is combining individual ratings over the set of items/alternatives captured from multiple criteria to make a group recommendation. For example, in a news recommendation system, individual ratings from different criteria for a set of news (e.g. importance, the relevance of location, and recency of news) could be aggregated to make a group recommendation [19]. By extending the strategy of aggregating personal preferences of group members, one can aggregate personal preferences on multiple criteria even for a personalized recommendation. The efficiency of group recommendation could be measured by how it satisfies the group members.

Group recommendation could be predicted by aggregating individual recommendations achieved by the collaborative filtering (CF) method [7]. In this approach, another perspective of group members' behavior can be concluded by comparing the individual recommendation with group recommendation.

By studying the behavior of people in social networks deeper, other factors such as the empathetic preference of people could be also considered in addition to their intrinsic preferences for group recommendation [42] [43]. In such settings, the group recommendation is computed through a weighted aggregation of preferences.

Group recommender systems could employ a group consensus function that would maximize the relevance of the item to the group and minimize the disagreement among the members simultaneously [8].

A simple method for achieving a consensus choice in the group is aggregating individual preferences. In addition, one can employ an algorithm to characterize the tendency of the group for achieving a more accurate group decision [9]. Seko et al. [9] propose a feature space for groups, and predict given scores by groups to the items. For each group, the feature space is expressed as the preference behavior of group members which is balanced (among group members) by observing the behavioral history of the group.

Gartrell et al. [10] proposed a method for recommending items to groups of people that not only considers the interest of group members but also focuses on social relationships among the members. By applying a consensus function that captures different types of social relationships between group members and characterizes the intended group, personal preferences/recommendations are aggregated to make the group recommendation.

With regard to more advanced problems, one can assume that only the preferences of some users in the group are observed. In this scenario, by utilizing the homophily and social influence in a social network, unobserved individual preferences could be learned and aggregated through the preference-oriented social network (POSN) model [11]. The POSN facilitates group recommendation even with incomplete preferences of group members.

Preference aggregation methods could be applied in the models that acquire the individual utility of group members. Two significant concepts in social utility (i.e. social welfare and fairness) could be utilized to develop a fairness-aware group recommendation framework [12]. Considering these individual utilities of group members is advantageous in making a balance between the preferences of the group members. In such a framework, for each group the concept of fairness-aware refers to maximizing the satisfaction for group members as well as minimizing the unfairness between them. Employing this fairness-aware model leads to higher group recommendation accuracy from individual preferences.

3.1.3.1 Preference Aggregation and Attention Mechanism

Inspiring by the methods mentioned in section 3.1.3, recently, there has been a growing interest in deploying attention mechanisms in deep learning for learning the aggregation strategies (i.e., group consensus functions) of group member's (predicted) preferences.

Learning weighted individual preferences to deploy group representations and learning group-item interaction has indicated promising results in this area. Cao et al. [13] proposed the attentive group recommendation (AGREE) model to solve the problem of preference aggregation for group recommendation, by learning the preference aggregation strategy from the data. They use the attention mechanism to reach the group representation, and then learn the interaction between groups and items. The most prominent contributions of this work are Learning groups' representations and group recommendations(learning the interaction between groups/users and items) by utilizing attention mechanism technique similar to Neural Collaborative filtering

(NCF) [44] (which is designed for personalized learning tasks). Existing strategies (mean of individual scores, least misery strategy, maximum satisfaction strategy) for aggregating individual preferences of group members are not flexible enough. Hence, in this work, they propose the attentive based mechanism for learning group representations from individual preferences such that it also considers the weight of preference for each user. For this purpose, an additive weighting function is applied on a set of users' representations to generate a target representation for groups. For generating a group representation as input of the model for learning the interactions, they sum up two representation vectors, one of them is a learned weighted individual preferences of group members and one for the learned group preference. Finally, for interaction learning, they learn user-item and group-item interactions at the same time which is then used for group recommendation and user recommendation.

The problem of aggregating individual preferences could be solved even for sparse data and cold start users by utilizing a model called Social Influence-based Group Representation (SIGR) [22]. The architecture of this model consists of a bipartite graph including two components: a social graph for learning the social influence of each user to achieve a group representation learning, and another one for learning user-item and groups-item interactions. In this work, the weight of users' preferences in different groups is captured and the attention mechanism is utilized for learning the users' social influence in the groups. Learning both user-item and group-item interactions using BGEM(bipartite graph embedding model) and learning social-influence based group representations based on aggregating individual preferences of group members are the main differences of SIGR from AGREE [13]. A bipartite graph is utilized for learning user-item and group-item interactions same as the general graph for learning user-user interactions (user social network). For learning group representations, a social graph of the users is constructed and network embedding approaches such as DeepWalk [45] and node2vec [46] are employed to get feature structure of each user. By assigning weights to different features, they extract social influence learning from those features. This leads to higher performance of SIGR Compared to AGREE, when data is sparse and we have cold start users.

Some side information could also be advantageous for increasing the efficiency of learning group representations from individuals, such as considering different sub-features like group description, external and internal social features and so on. by utilizing a multi attention-based group recommendation model (MAGRM) [23]. The group representation vector would be generated by concatenating the sub-features vectors. Finally, a neural attention mechanism is employed for learning preferences of groups on items (learning group-item interactions).

The main contribution of this thesis which makes it distinguishable from existing work in this area is that most of the these studies assume that different user-item, groups-item, and user-user interactions (including preferences of groups and users) are observed and based on that learn group representation and predict the top choice of the group, while our main assumption is that we have just observed groups' top choices.

Group recommender systems can play the role of mediator for group members' preferences, such that the recommended item would satisfy all the group members. In fact, rather than focusing on the type of users' preferences to be conflicting in the groups, a fair procedure for aggregating individual preferences could be utilized [47].

The performance of described methods in this section strongly depends on the type of the items, type of the study environment, available information in the dataset, and the objective goal of the recommender system model. Under different criteria, each strategy would reveal better performance. Moreover, different parameters in the system could affect the performance of the group recommendation model. Some significant parameters affecting the efficiency of group recommender systems could be considered as the size of the groups and the strategy/algorithm of recommendation [48].

3.2 Social Choice and Preference Learning

Social choice equips group recommender systems with a principled framework for aggregating individuals' preferences into group preferences (or decisions). The proficiency of the social choice theory could be deliberated in real-world scenarios from sharing food among groups of friends to voting in an election or a committee. Many social choice schemes (such as voting rules) have been studied so far [20].

The problem of preference assessment in social choice settings is very predominant as a collection of user preferences must be elicited or learned to make a group decision. The research has focused on two approaches of *preference elicitation* via asking relevant queries which result in revealing user preferences and *preference learning* from historical user-item interaction data.

3.2.1 Preference Elicitation

In preference elicitation/manipulation methods the key idea is step by step asking simple queries about the users' preferences instead of having their preferences record straightway.

The preference elicitation methods for social choice problems has been developed based on diverse approaches ranging from heuristic methods using the concept of *possible and necessary winners* to the notion of *minimax regret*.

3.2.1.1 Possible and Necessary Winners

Given a profile of partial orders and a candidate item, necessary and possible winners refer to the guarantee for the candidate item to win and the possibility of winning for it respectively [15]. Solving these problems mostly relies on the choice of voting rules. While there is some related work in which choosing the voting rule is an uncertain process [49], Xia and Conitzer [15] consider a fixed voting rule to investigate the possible/necessary winner candidate.

In these problems, the partial order of voters could be extended to complete ranking orders. Possible winners in this problem can be inferred as candidate winners in some of the complete extensions of ranking order, while necessary winners refer to winners in all of the complete extensions [50].

Considering a fixed voting rule, in a score-based voting rule like the Borda decision rule, possible winners in each group could be assumed as the candidates that their maximum score is greater than the minimum score of all the other group members. While the necessary winners are the candidates that their minimum score is inferred greater than the maximum score among all the group members in each group [14].

3.2.1.2 Minimax Regret

The minimax regret method is useful in scenarios that aggregating personal preferences and making decisions are uncertain. In other words, users are faced with uncertain states and multiple items that could be selected. Regret could be defined as maximum opportunity loss values which are measured by the users for each item (alternative) during making decisions. The concept of minimax regret refers to choosing the item (alternative) with the lowest maximum regret by the users. This technique has been used to increase the power of the decision-making process and reach preference elicitation in different domains.

Deploying minimax regret has indicated promising results in tackling existing problems related to preference elicitation, from deriving the best practicable outcome in the scenarios in which the utilities of users are not specified precisely, to deploying practical elicitation methods for reducing the utility uncertainty to achieve an optimal decision with minimum interactions of users [51].

Assessing the power of the minimax regret approach in making efficient recommendations has been studied over users in a recommender system for helping students to navigate and find rental accommodation [52].

Defining the class of minimax regret in mechanism design for partial revelation instead of full utility revelation from agents could achieve approximate dominant strategy implementation [53].

Considering incomplete voter preferences, Lu and Boutilier [16] measure the minimax regret for several prominent voting rules. Rather than selecting necessary/possible winners, they propose a technique in which the item (alternative) that minimizes maximum regret (that compute the worst-case error) would be selected as the actual winner. The robustness of their optimization is demonstrated along with the efficiency of regret-based-elicitation in selecting approximate and even exact winners.

By extending the problem of single-winner to the problem of selecting a poll with multiple items (alternatives) by adopting minimax regret approach Lu and Boutilier [54] investigate the problem of proportional representation by applying positional scoring functions considering incomplete (partial) preferences of voters.

3.2.2 Preference Learning

There is also growing interest in predictive models of preferences for learning user preferences in social choice settings.

Hughes et al. [17] proposed efficient algorithms to drive the optimal Bayesian approach which is applied widely in social choice problems that aim to compute an optimal decision based on the voters' preferences.

Preference-oriented social network [11] would capture the correlations among people who have interactions with each other in the social network. After that, the model has the ability to learn and predict the unobserved preferences of users (in the form of rankings over the set of items) in the network which leads to making efficient group recommendations in a social choice context.

To tackle the issue of partial preferences of voters in the system, an application of machine learning methods is proposed to predict the missing preferences [18]. Partial rankings problem in voting systems indicates partial ordering over preferences provided by the voters. The proposed application would learn the preferences to predict the missing components using latent patterns in the provided information to prepare them for voting rules dealing with missing data.

3.3 Recommender Systems and Neural Networks

Deep learning has recently shown enormous potential in improving the performance of recommender systems by capturing enriched user and item representations [55]. This is mostly due to the ability of deep learning in capturing the non-linear relationships between users and items. It also has a strong performance in learning complex representations of the input data through the hidden layers.

So far, different types of neural networks are successfully applied for the top-k recommendation. There exist different architecture for deep neural networks such that each of them is suitable for tackling specific types of problems.

3.3.1 Multilayer perceptron (MLP)

Multilayer perceptron (MLP) or feedforward networks which are extensively described in section 2.2.2, is our intended architecture for our proposed group recommender system model.

Some recent work has developed deep learning models for a variety of recommendation tasks. It is performed extensively in mapping users and items to a latent space to measure the similarity between them in a content-based recommender system model for modeling cross-domain behavior of users [56].

Deep feedforward networks are utilized in learning unseen feature combinations via low-dimensional dense embeddings (vector representations) for sparse input features [57]. To tackle the input sparsity problem which leads to over-generalizing in deep neural networks and causes recommending less relevant items to the users, Cheng et al. [57] jointly trained wide linear models and deep neural networks for sparse and high-rank user-item interactions. The deep component of their proposed model is a multilayer perceptron (MLP) that takes the feature strings (for categorical features) as the input. These features string is converted into low-dimensional and dense real-valued vector representation (embedding), which is then fed through the hidden layers for training the model.

Rather than focusing on learning the auxiliary information of users or items, modeling the user-item interactions from implicit feedback data has established more valuable work in this area [44]. He et al. [44] aim to address the problem of collaborative filtering with implicit feedback data by proposing a deep neural network architecture to model latent features of users and items. They call they represented generative framework neural collaborative filtering (NCF), which reveals the collaborative filtering-based recommendation using neural networks.

3.3.2 Convolutional Neural Networks

Convolutional neural networks are used in a variety of recommendation tasks such as automatic music recommendation. For modeling acoustic features of music, by learning the representations of the audio signals with deep convolutional neural networks, the latent factors from music audio could be predicted (in case of cold start problem that these factors are not available in the usage data) to overcome the existing limitations in collaborative filtering (CF) approaches [58].

Matrix factorization (MF) is one of the basic methods used in collaborative filtering models. Another model is LightFM, a hybrid content-collaborative model, represented to tackle the problem of cold-start users and items in simple collaborative filtering (CF) and content-based methods [59]. The main idea is that this model would learn the representation of users and items required for matrix factorization (MF) as linear combinations of their content features' latent factors. Similar to other collaborative filtering methods, this model uses the users and items latent vector representation (embedding) for the recommendation. The difference is that it applies a function to combine the content features (which would describe each item or user) linearly. It utilizes image-based convolutional neural networks (CNN) for modeling the item content information. This model has the ability to learn the representation of users and items based on the provided interaction data between them on one hand, and predict the ratings for new users and items on the other hand.

3.3.3 Recurrent Neural Networks

Recurrent neural networks (RNN), are advantageous in real-world scenarios in which the profile of users or features of the items are dynamic. More precisely, in such situations, features of items or interest of users may change over time. In such cases, deploying a recurrent neural network (RNN) could result in an efficient recommendation for modeling temporal evolution [60].

3.3.4 Autoencoders

Classical autoencoders (AEs) are used in the architecture of a joint collaborative autoencoder framework that learns user-user and item-item correlations jointly (simultaneously) [61]. This model is a neural neighborhood-based collaborative filtering (CF) approach that applies a pairwise hinge-based objective function to optimize the top-K recommendation problem. In this way, the recommender system can predict more accurate recommendations due to taking advantage of hidden information.

Zhang et al. [62] proposed Collaborative Knowledge Base Embedding (CKE) which is considered as a heterogeneous network that contains structured/unstructured data from all the users and items. For generating the item latent vector in the collaborative filtering (CF) method, they concatenate four different vectors. The first vector indicates the interaction between items and users and captures the implicit feedback of users to items. The second vector could be considered as the structural embedding. For learning this vector, they have considered structural knowledge, which is a network that contains all the information of users, items, their features, and the relations between all of them. To generate its corresponding vector, they have assumed this network as a graph and used TransR (an existing method to convert the graph to vector). Two other vectors are generated considering textual knowledge and visual knowledge indicating the content information of items and the images related to the item respectively. For learning these vector representations they have used autoencoders (AEs) appropriate to the type of the data. One noticeable point in this work is the ability to learning feature embedding for users.

3.3.4.1 Denoising Autoencoders

Employing denoising autoencoders could be beneficial for addressing the top-k recommendation problem [63]. The main contribution of the proposed model is considering the presumption of observing corrupted user-item from the complete version of users' preference collection. The model has the capability of learning latent representations for corrupted user-item preferences which can generate the complete input again. Specifically, a subset of an item preference set for a user is fed into the model to recover the complete item set during training. During testing (predicting) time, given the existing preference set as input, the model would predict new items to the user. In this way, training on corrupted data has indicated immense performance for collaborative filtering methods.

The collaborative filtering (CF) method could be employed jointly with learning deep representation of items' content information to recommend items to users [64]. In this paper, to gain a better performance when the auxiliary information is sparse, they represent a deep learning model CDL (collaborative deep learning) for learning the representation of items more effectively. They utilize a Stacked Denoising Auto Encoder (SDAE) which is a feedforward neural network to learn the representation of the input data. Specifically, a generalized Bayesian SDAE is employed to learn the latent vector representation for items. Simultaneously, the collaborative filtering (CF) based model will predict the ratings for items, given the latent users and items vector representations. In this approach, for learning a more powerful representation of items' content, the context could be learned more precisely by considering the bag-of-words method.

3.3.4.2 Variational Autoencoders

Variational autoencoders (VAEs) are non-linear probabilistic architectures for deep neural networks that have shown an enormous advantage in designing recommender systems. Sparsity and cold start problems are some existing limitations in collaborative filtering (CF) methods. Moreover, assuming only text modality of the content may result in learning poor representations. To address these two issues, a Bayesian generative model which is called collaborative variational autoencoder (CVAE) is proposed for considering rating and content jointly for the recommendation in multimedia scenarios [65]. In an unsupervised learning task, the deep network is not only able to learn latent representations from content data, but also it would learn the implicit user-item relationships between both content and rating.

Another example of extending variational autoencoders (VAEs) to collaborative filtering from implicit feedback is a neural generative model with multinomial conditional likelihood [66]. It benefits from the ability of VAEs to generalize linear latent factor models to non-linear probabilistic latent variable models. This is mainly due to the power of deep neural networks on large and sparse datasets provided for recommendation tasks.

Joint variational autoencoder (JoVA) [67] which is consists of two variational autoencoders (VAEs), is a variational autoencoder based collaborative filtering (CF) model for a top-k recommendation from implicit feedback data. The two components would jointly learn user and item representations by capturing user-user and item-item correlations. The model could learn both users and items representations along with modeling their uncertainty and would reconstruct item representations and predict user preferences.

3.4 How This Thesis Fits

In this chapter, we explored a variety of research in the area of group recommender systems, social choice theory, preference learning, and recommender systems and deep neural networks. Studying these concepts would be advantageous in understanding the basic concepts of our research statement in this thesis.

For a group recommendation, all the mentioned methods in sections 3.1 assume that user preferences/profiles of group members or group preferences are accessible.

Our goal differentiates from this body of research. We make a group recommendation to a new group of users whose preferences (or profiles) were not observed, but their participation in group decision making of some other groups has been observed.

In section 3.2 we focused on studying existing research in the area of social choice theory from the basic usage of this concept to deploying it in more complex learning networks. However, in this thesis, our work differs in several ways. Rather than eliciting or inferring missing user preferences for group decision making, we predict the group decision (or preferences) from some other groups' observed decisions. A special instance of our problem, reverse social choice, has the opposite goal of social choice by segregating user preferences from group decisions.

The variety of models mentioned in section 3.3 cannot directly be applied to our group recommendation problem with implicit feedback, as their emphasis is on predicting user preferences from user-item interactions.

While our proposed model would address the top-k group recommendation problem by utilizing deep neural networks, it has novel aspects that differ from existing work. In the next chapter, we describe our proposed approach in more detail.

Chapter 4

Approach

In this chapter, we define the problem we aim to address in this thesis which is making group recommendation from implicit feedback. We further consider two specific instances of the general problem that we are interested in. Following by problem definition, we represent our approach for tackling the problem and introduce our proposed DeepGroup model and investigate its architecture in detail.

4.1 Group Recommendation from Group Implicit Feedback

Our goal is to make a recommendation to a group of users whose personal preferences are unknown to the system. However, these users might have participated in group decision making processes of some other groups with known/public decision outcomes. Since we assume the decision histories of the groups, the input data for our proposed problem is considered as the group implicit feedback data.

The applications of this problem are prevalent in our daily lives. For instance, recommendations of restaurants or vacation packages to a group, when we have observed restaurants or places which they have visited with some other friends or family members.

We consider a set of n users $\mathcal{U} = \{1, \dots, n\}$ and a set of m alternatives (items) $\mathcal{A} = \{a_1, \dots, a_m\}$. We assume that we have observed l groups of users $\mathcal{G} = \{G_1, \dots, G_l\}$ with $G_i \subseteq \mathcal{U}$, and their corresponding group decisions (choices). The observed group decisions can be represented as the *group-item interaction matrix* $\mathbf{Y} = [y_{ij}] \in \{0,1\}^{l \times m}$, where $y_{ij} = 1$ if $G_i \in \mathcal{G}$ has decided $a_j \in \mathcal{A}$ as its group decision (positive samples). In this setting, one can focus on the top-k recommendation problem by suggesting the k most preferred (or likely) items from $\mathcal{A}$ to a new group of users $G \subseteq \mathcal{U}$ where $G \notin \mathcal{G}$. Of course, one can extend our problem to the setting in which observed decisions/outcomes are in the form of group aggregated rankings rather than a consensus option.

While our defined problem covers a broad range of problems, of particular interest, is two special instances of this problem:

Group Decision Prediction. Single-option group recommendation (i.e., when $k = 1$)—sometimes referred to as *group decision prediction*—not only applies to group recommendation, but also can be used for predicting the most likely decision (or outcome) of a newly formed group G. Imagine a committee is asked to decide on a sensitive issue (e.g., promotion, social injustice issues, etc.) when various decisions are possible. The goal is to predict the final decision of the committee based on the involvement of the committee members in previous committees whose final decisions are public.

Reverse Social Choice. By letting the target group G be a singleton set of a user $u \in U$, one can focus on a special instance of our problem, that we call *reverse social choice*. As opposed to social choice functions that aggregate individuals' preferences to group decisions or group preferences, the reverse social choice intends to map group decisions to individuals' preferences. The solution to this problem not only helps us to enhance preference learning but also allows us to measure privacy leakage from publicly announced group decisions.

Regardless of our interest in these two special instances of the group recommendation problem, our proposed solution is for the general problem and applicable to any instances. Our general approach is to predict the likelihood of the interaction of group G with any item in $\mathcal{A}$ (or preference of group G over $\mathcal{A}$), and then select a rank list of k items with the highest prediction score for recommendation to the group G. Our learning task in this paper is to find the *likelihood function* $f(G, a|\boldsymbol{\theta})$ that predicts the likelihood of group G's interaction with any item $a \in \mathcal{A}$. Here, $\boldsymbol{\theta}$ denotes the model parameters and can be learned (or estimated) from the observed groups $\mathcal{G}$ and group-item interaction matrix $\mathbf{Y}$. Our proposed model, DeepGroup, is described in Section 4.2 as a powerful deep learning model for formulating and learning this likelihood function.

4.2 DeepGroup Model

We propose DeepGroup neural network to address the group recommendation problem discussed in Section 4.1. DeepGroup, by learning the likelihood function $f(G, a|\boldsymbol{\theta}) : 2^U \times A \to [0, 1]$ for $G \subseteq U$ and any $a \in A$, can be used to predict the likelihood of group G_i's interaction with any item $a_j \in A$ by $\hat{y}_{ij} = f(G_i, a_j|\boldsymbol{\theta})$. We discuss the model architecture of DeepGroup, its key components (e.g., aggregator functions), and its learning process in this Section.

4.2.1 Model Architecture

Figure 4.1 depicts the architecture of DeepGroup. The DeepGroup model takes both group $G_i \subseteq U$ and item $a_j \in A$ as an input. The G_i is represented as the n-row sparse vector $\mathbf{g} = [g_p]$ where $g_p = 1$ if $p \in G_i$ otherwise $g_p = 0$. The item a_j can be represented by one-hot encoding in m-row vector $\mathbf{a}$. Each primitive input of DeepGroup is a group (including the group members indexes), and an item (including its index) in the set of items.

The DeepGroup considers low-dimensional real-valued latent representations (or embedding which are initialized randomly) for all users $u \in U$ and items $a \in A$. The latent representations of users and items are captured by $n \times d$ matrix $\mathbf{U}$ and $m \times d'$ matrix $\mathbf{V}$ (resp.), where d and d' are the dimensions of user and item latent spaces (resp.). For the input group $\mathbf{g}$, DeepGroup retrieves all its users' latent representations $\{\mathbf{U}_p | p \in U \text{ and } g_p = 1\}$, where $\mathbf{U}_p$ denotes latent vector of user p (i.e., the p^{th} row in the matrix $\mathbf{U}$). Similarly, DeepGroup looks up the item embedding $\mathbf{V_j}$ for input item a_j.

A key idea behind DeepGroup is the aggregation of a group $\mathbf{g}$'s users latent representations $\{\mathbf{U}_p | p \in U \text{ and } g_p = 1\}$ into a single fixed-length vector $\mathbf{q}$:

$$\mathbf{q} = \text{Aggregate}\left(\{\mathbf{U}_p | p \in U \text{ and } g_p = 1\}\right). \tag{4.1}$$

The Aggregate(.) function takes any set of user latent representations and maps them into $\mathbf{q}$, which is the latent representation of the group $\mathbf{g}$. This group latent representation is expected to capture the consensus preference of group members. We discuss about different aggregator functions in Section 4.2.2.

The group latent representation $\mathbf{q}$ and the item embedding $\mathbf{V}_j$ are then concatenated and fed into a multilayer perceptron (MLP) neural network to predict $\hat{y}_{ij}$ (i.e., the likelihood that group g decide on item a_j). The MLP consists of X hidden layers formulated by

$$\begin{aligned} \mathbf{h}^{(1)} &= g^{(1)}\left(\mathbf{W}^{(1)}\left(\mathbf{q} \parallel \mathbf{V}_j\right) + \mathbf{b}^{(1)}\right) \\ \mathbf{h}^{(i)} &= g^{(i)}\left(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}\right) && \text{for i} > 1, \end{aligned}$$

where $\mathbf{h}^{(i)}$, $\mathbf{W}^{(i)}$, $\mathbf{b}^{(i)}$, and $g^{(i)}(.)$ are the hidden unit vector, the linear transformation weight matrix, bias vector, and non-linear activation function for layer i, respectively. Here, $\parallel$ is the concatenation operator. Finally, the output layer of DeepGroup computes $\hat{y}_{ij}$ by

$$\hat{y}_{ij} = \sigma\left(\mathbf{w}_o\mathbf{h}^{(X)} + b_o\right), \tag{4.2}$$

where $\sigma(.)$ is the sigmoid function for converging the linear transformation of the last hidden layer output $\mathbf{h}^{(X)}$ into a probability. Here, $\mathbf{w}_o$ and b_o are the weight vector and bias parameter for the output layer.

4.2.2 Aggregator Functions

An integral part of DeepGroup is the aggregate function which maps any arbitrary set of user embeddings into group representation $\mathbf{q}$. A candidate function is required to satisfy at least two natural properties.

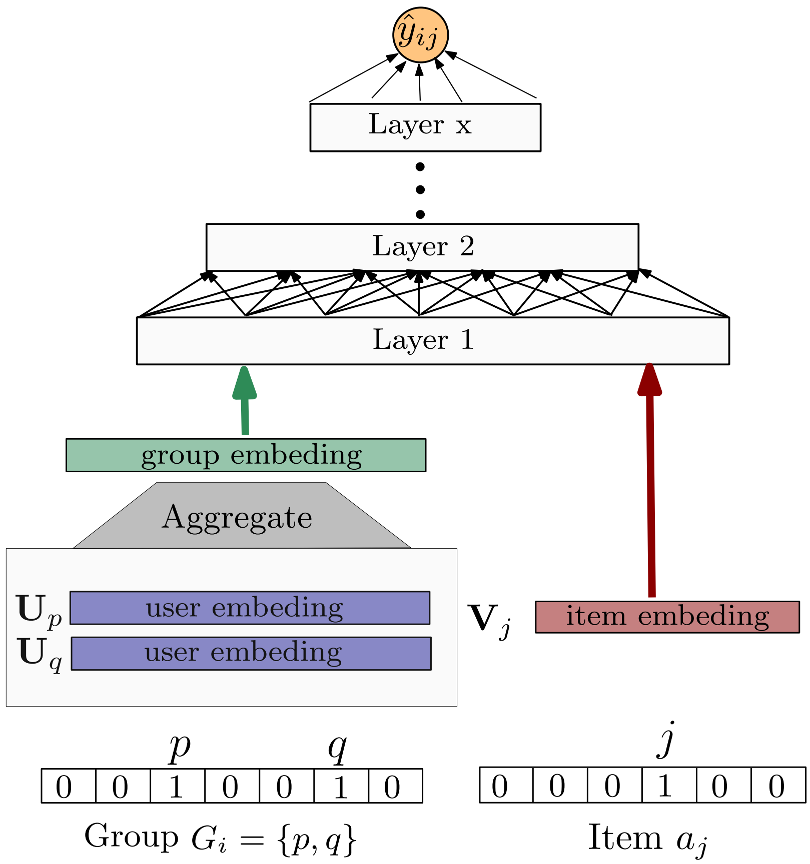


Figure 4.1: The architecture of the DeepGroup model.

Property 1 *A function* Aggregate *acting on sets of user embeddings must be* ***permutation invariant*** *to the order of user embeddings in the set such that for any permutation* π *and any user embedding set* $\{\mathbf{U}_{i_1}, \cdots, \mathbf{U}_{i_j}\}$*:*

$$\text{Aggregate}\left(\{\mathbf{U}_{i_1}, \cdots, \mathbf{U}_{i_j}\}\right) = \text{Aggregate}\left(\{\mathbf{U}_{\pi(i_1)}, \cdots, \mathbf{U}_{\pi(i_j)}\}\right).$$

Property 2 *A function* Aggregate *acting on sets of user embeddings must have a* ***fixed-length range*** *for any set of user embeddings. In other words, letting* $\mathcal{E} = \{\mathbf{U}_i | i \in \mathcal{U}\}$ *be the set of all users' embeddings, the function* $Agg : 2^{\mathcal{E}} \rightarrow \mathbb{R}^k$ *maps any* *subset of $\mathcal{E}$ to a fixed-length real-valued vector with the dimensionality of k.*

Given these two properties, one can consider two classes of aggregate functions.

ELementwise Aggregators. By deploying an elementwise operator (e.g., mean, max, min, median), an elementwise aggregator reduces a group G's user embeddings $\{\mathbf{U}_p|p \in G\}$ into a group embedding $\mathbf{q}$. This class of aggregators generates the group embedding with the same dimensionality of user embedding. An important instance of this class is the Mean aggregator which compute the i^{th} element of the group embedding $\mathbf{q}$ by:

$$q_i = mean(\{u_{pi}|p \in G\}), \tag{4.3}$$

where u_{pi} denotes the value of the i^{th} dimension of user p's embedding. A variant of this Mean aggregator has been widely deployed in representation learning on graphs convolutions network to aggregate features from a node's neighborhood. For example, one of the existing strong models is GraphSAGE [68] which will utilize the mean aggregator function for node's features aggregation in a graph for generating embeddings for a node that its data is unobserved. Similar to our case of study, in which we aim to generate group embedding from user embeddings, in this approach, generating a specific node representation from its neighborhood nodes requires a symmetric aggregation function that is not in accord with the ordering of the nodes in the neighborhood.

By replacing mean in Eq. 4.3 to any other elementwise operators (e.g., min, max, median, etc.), one can derive other elementwise aggregator functions.

Combined Aggregators. While an elementwise aggregator (e.g., Mean aggregator) can reduce a set of user embeddings into a single group embedding, it is possible that two distinct sets of user embeddings result in the same group embedding under a specific aggregator. To address such issues and make group representations more distinctive, one can combine multiple aggregators by concatenating their outputs. For example, we can define the mean-max-min aggregator for aggregating a set of user embeddings $\mathbf{U}_G = \{\mathbf{U}_p | p \in G\}$ by

$$MMM(\mathbf{U}_G) = Mean(\mathbf{U}_G) \parallel Max(\mathbf{U}_G) \parallel Min(\mathbf{U}_G), \tag{4.4}$$

where $\parallel$ is the concatenation operator, and Mean, Max and Min, are elementwise aggregator functions. This combined aggregator has a fixed-length range with three times bigger dimensionality of user embeddings. The mean-max-min aggregator has an interesting geometric characteristic. While the Min aggregator function can capture the lowest impact of user embedding features, the Max aggregator considers the most severe features. On the other hand, the Mean aggregator function deploys an average combination over these features. The min and max aggregators return two farthest corners of the minimum bounding box specified by the set of user embeddings $\mathbf{U}_G$.

4.2.3 Learning DeepGroup

One can learn DeepGroup model parameters by the *maximum likelihood estimation (MLE)* method. Given the observed groups $\mathcal{G}$ and group-item interaction matrix Y, the log likelihood can be computed by

$$\ell(\boldsymbol{\theta}|\mathcal{G}, \mathbf{Y}) = \sum_{i=1}^{l} \sum_{j=1}^{m} y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log (1 - \hat{y}_{ij}), \tag{4.5}$$

where $\hat{y}_{ij} = f(G_i, a_j|\boldsymbol{\theta})$ is DeepGroup's estimated probability for interaction of group G_i with the item a_j. The maximum likelihood estimate of the model parameters are

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}|\mathcal{G}, \mathbf{Y}). \tag{4.6}$$

Equivalently, one can learn model parameters by minimizing following loss function:

$$L(\boldsymbol{\theta}|\mathcal{G}, \mathbf{Y}) = -\ell(\boldsymbol{\theta}|\mathcal{G}, \mathbf{Y}). \tag{4.7}$$

This loss function is the same as the *binary cross-entropy loss*, which can be minimized by performing *stochastic gradient descent (SGD)* or any other optimization techniques. While deploying the architecture of DeepGroup, this loss function can be modified to or extended with a *pairwise loss* [69], which directly optimizes the ranking of the group-item interactions.

Chapter 5

Experiments

We conduct extensive experiments to evaluate the effectiveness of our proposed DeepGroup model for both problems of group decision prediction and reverse social choice described in Section 4.1.[1] We define three datasets for our experiments, investigate the process of group generation, and define the proposed benchmarks that we compare the prediction accuracy of DeepGroup to them. Finally, we describe the empirical results of our experiments. Specifically, we investigate the efficiency of DeepGroup for different group generation methods, different group decision rules, and different aggregator functions for both group decision prediction and social reverse choice tasks in all the preference datasets.

5.1 Group Datasets

The group datasets should consist of both group membership data (i.e., group structures) $\mathcal{G}$, and group decisions in the form of group-item interaction matrices $\mathbf{Y}$. Due

[1] `https://github.com/sarinasajadi/DeepGroup`

to inaccessibility to such group datasets, we create our group datasets using real-world preference datasets, different group formation mechanisms, and group decision rules (or voting methods).

Real-world preference datasets. We consider four real-world preference ranking datasets. Three datasets are from the 2002 Irish Election:[2] Dublin West with 9 candidates and 29,989 user preferences; Dublin North containing 43,942 user preferences over 12 candidates; and Meath containing 64,081 user preferences over 14 candidates. The user preferences in these datasets are partial rankings of the top-t form (i.e., the ranking of the t most preferred candidates). Our other dataset is the Sushi dataset consisting of 5000 user preferences as complete rankings over 10 varieties of sushi.[3] For converting implicit group datasets from these individual ranking datasets, we describe the process of making groups of users and generating group decisions in the following sections.

Group generation methods. To generate a set of groups $\mathcal{G}$ from real-world preference datasets, we deploy various methods. The *κ-participation group (KPG)* method first samples n users from a preference dataset, then κ times randomly partitions this set of users into size-constrained subsets (i.e., groups), whose sizes are bounded by $[s_{min}, s_{max}]$. The KPG outputs the collection of all unique subsets generated by these κ partitions. By varying κ, one can control the extent to which each user has participated in different groups (or equivalently, the extent to which groups overlap with one another). The algorithm of this method is described in Algorithm 1.

In another scenario, we aim to study the effect of homophily and heterophily in

[2] `http://www.preflib.org/data/election/irish/`
[3] `http://www.preflib.org/data/election/sushi/`

Algorithm 1 Generating groups with different κ-participation values

D_0 = Uniformly randomly choose n users from D without replacement
$D_{final} = \{\}$
for k times **do**
    $D' = D_0$
    while D' is not empty **do**
        select g_s uniformly randomly from $[s_{min}, s_{max}]$
        G = Select Min(g_s, $|D'|$) users uniformly randomly from D'
        $D' = D'$ - G
        $D_{final} = D_{final} \bigcup \{G\}$
return D_{final}

the groups. The *Random Similar Groups (RSG)* method randomly selects l groups from a preference dataset, where the group size is randomly picked from $[s_{min}, s_{max}]$ and group members have similar preferences. The preference similarity is enforced by ensuring that all pairwise Kendall-τ correlations of group members are at least τ_{sim}. *Random Dissimilar Groups (RDG)* method has the similar stochastic process of RSG with the difference that group members must have dissimilar preferences. The dissimilarity is imposed by ensuring that all pairwise Kendall-τ correlations of group members are at most τ_{dis}. RSG and RDG can be easily implemented by rejection sampling.

In our experiments, we set $s_{min} = 2$, $s_{max} = 10$, $\tau_{sim} = 0.5$, and $\tau_{dis} = -0.5$ while varying other parameters.

Group decision rules. We create the group-item interaction matrix $\mathbf{Y}$ for each generated group set $\mathcal{G}$ by voting rules [20]. To do so, we aggregate user preferences of each group $G_i \in \mathcal{G}$ to a group decision $a_j \in \mathcal{A}$. We focus on Borda and plurality—two examples of positional scoring rules—in which an alternative a, for each preference ranking r, receives a score $g(a, r)$ based on its ranking position. Then, the group decision is the alternative with the highest cumulative score over all rankings of group members.

The Borda score function is $g_B(a, r) = m - r(a)$ and the plurality score function is $g_P(a, r) = \mathbb{1}[r(a) = 1]$ where $\mathbb{1}[.]$ is an indicator function, and $r(a)$ represents the position of a in the ranking r. In our experiments, for a fixed group set $\mathcal{G}$, we either use Borda for all $G_i \in \mathcal{G}$, plurality for all $G_i \in \mathcal{G}$, or uniformly at random select between Borda and plurality (i.e., mixture of Borda and Plurality). Borda treats utility differences as linear, whereas plurality utility is "all or nothing." We also note that plurality has been widely-used in many group decision settings (including elections) and Borda is a useful surrogate for random utility models [70].

5.2 Benchmarks

We compare the prediction power of DeepGroup against some baseline algorithms. Since we are not aware of any other solutions to our problem, we have designed these heuristics benchmarks. These baselines intended to provide some benchmarks for our proposed solution, not necessarily prove its superiority over all possible solutions and heuristics. We have also been cautious in our conclusions to focus more on other contributions rather than the performance of our solution.

- **Popularity (Pop)** predicts the most popular group decisions in the training set as the group decision of any groups in the testing set.

- **Random Top Choice Plurality (RTCP).** For a group in the testing set, RTCP first guesses its users' top choices (or votes), then outputs the plurality winner (ties broken randomly). To guess user top choice, if a user belongs to at least one group in the training set, RTCP randomly picks one of its groups and assign that group's decision as its top choice. For those users who don't belong to any groups in the training set, the method guesses their top choices to be the popular group decision (or item) in the training set.

- **Overlap Similarity (O-Sim).** For a given group in the testing dataset, this method outputs the group decision of the most similar group in the training set, when the similarity is measured by the number of common members.

5.3 Experimental Setup

In all our experiments, DeepGroup has four hidden layers (i.e., X=4) with 64, 32, 16, and 8 hidden units and the Relu activation function. To prevent overfitting, we use dropout over hidden layers with a probability of 0.8 for retaining each node in a hidden layer. The user and item embedding dimensions are both set to 64 and the Mean aggregator is used (unless noted otherwise). We optimize DeepGroup with Adam optimizer for 100 epochs with a learning rate of 0.001 and the batch size of 4096 (i.e., each mini-batch encompasses 4096 records of negative/positive group-user interactions along with group membership data).

For both group decision prediction and reverse social choice tasks, we compare the performance of DeepGroup and baselines by their prediction accuracy, measuring the percentage of correct top-choice prediction (i.e. group decision) for groups in the testing set.

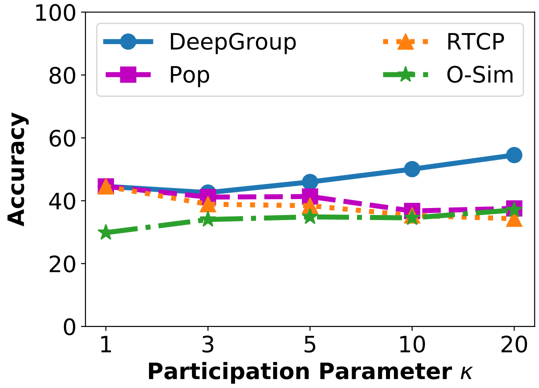
As our group dataset generation is stochastic, for each fixed setting (e.g., preference dataset, group set generation, group decision rule, etc.), we generate 20 instances of each group dataset setting and report our results as an average accuracy over those instances. In experiments focused on group decision prediction task, for each group dataset, we randomly selected 70% of all groups and their group-item interactions as the training set and 30% for the testing set. The groups in the testing sets are not observed in the training set, but their members might have appeared in some other groups in training. The only constraint regarding the input data of DeepGroup is when there are cold-start users in the testing set. In such scenarios, the model should be capable to predict the user representation vectors of cold-start users accurately. For this purpose, DeepGroup sets the user embedding of a new user in the testing set with the average of learned user embeddings, thus assuming the average or *default* preference. When the task is the reverse social choice, we use each group dataset as the training set and create a testing set including the singleton groups of all users that appeared in the groups of the training set.

5.4 Empirical Results

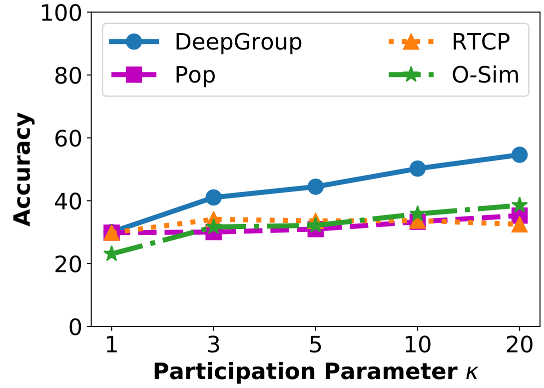
We report the empirical results of our extensive experiments for both group decision prediction and reverse social choice in different settings.

5.4.1 Group Decision Prediction on κ-Participation Groups

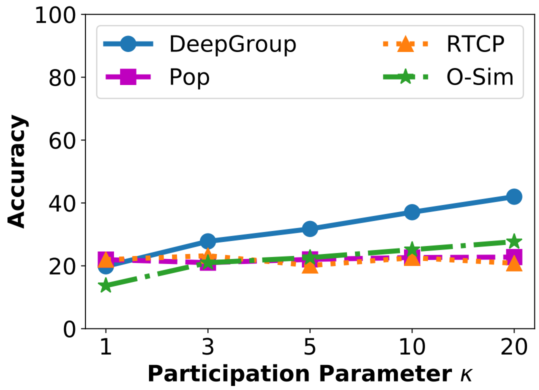
We aim to investigate the effectiveness of DeepGroup in group decision prediction when compared to other benchmarks under various group datasets generated with real-world preference data and κ-participation group generation. We fix the group decision rule to plurality or Borda for all generated groups. By fixing the number of



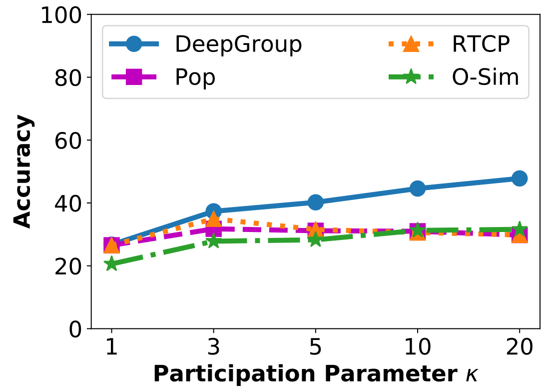
(a) Sushi



(b) Dublin West



(c) Dublin North

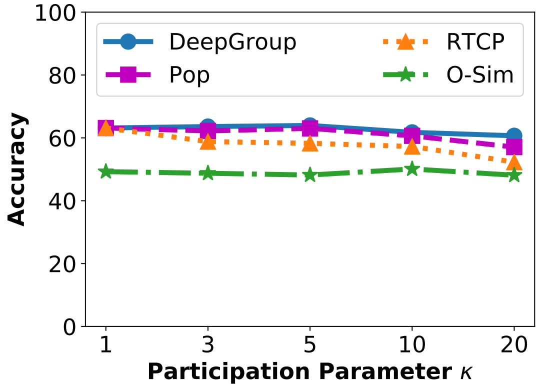


(d) Meath

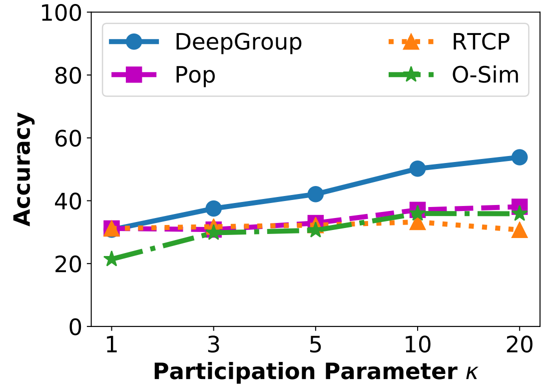
Figure 5.1: The accuracy of DeepGroup and other benchmarks for different group datasets generated on various preference datasets (a)–(d) with κ-participation method, and the plurality group decision rule.

users $n = 5000$ and varying κ over $\{1, 3, 5, 10, 20\}$, we study how the performance of different methods change with more availability of implicit data (i.e., the participation of individuals in different group decisions).

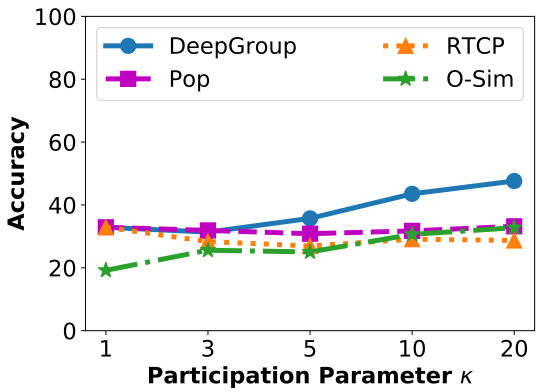
Figures 5.1 and 5.2 show the accuracy of different methods for various group datasets for the plurality and Borda decision rules respectively. In all four datasets, DeepGroup performs comparably with benchmarks for $\kappa = 1$ but outperforms the



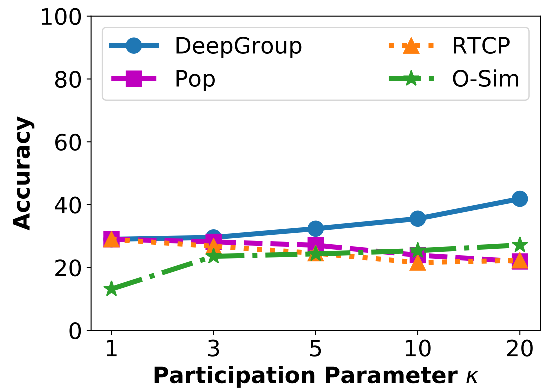
(a) Sushi



(b) Dublin West



(c) Dublin North



(d) Meath

Figure 5.2: The accuracy of DeepGroup and other benchmarks for different group datasets generated on various preference datasets (a)–(d) with κ-participation method, and the Borda group decision rule.

benchmarks for $\kappa \geq 3$. The performance of DeepGroup is more prominent as κ increases (e.g., about 100% improvement over the best baseline for $\kappa = 20$ and Irish datasets).

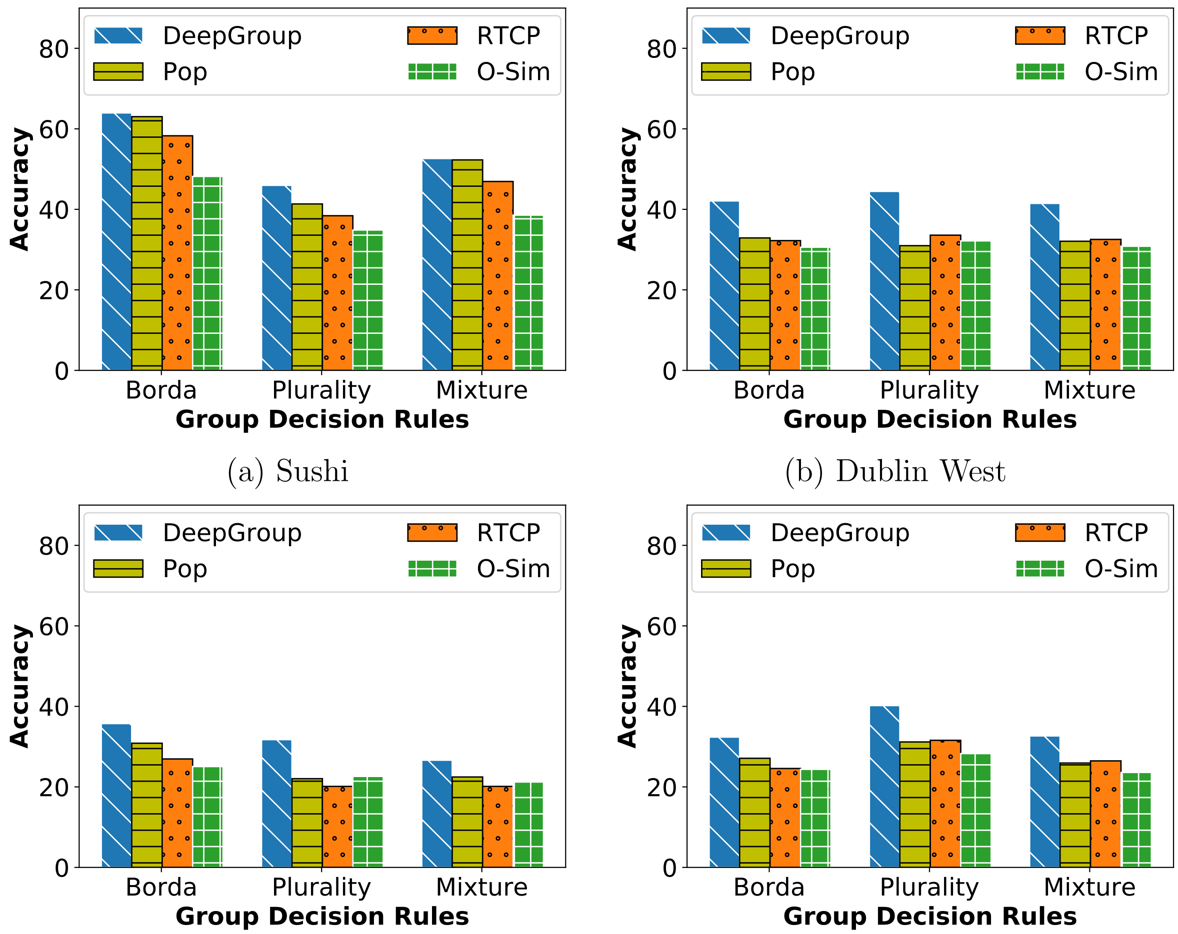
As we observe in Figures 5.1 and 5.2, for sushi dataset, the Borda decision rule indicates higher prediction (more than 60% for DeepGroup and most of baselines). For the Borda group decision rule and sushi dataset, our model performs similar to the benchmarks and have a steady behavior when we increase κ. However, its accuracy is still a little higher than the benchmarks for $\kappa \geq 3$. These behaviors of our model on the Sushi dataset could be mainly due to the lowest size of users datasets in Sushi and an imbalanced ranking preference in this dataset (there is one specific type of Sushi that is more popular among the users). The latter reason can explain the closeness of DeepGroup predictions to other benchmarks when the group decision rule is Borda. on the other hand, when the plurality group decision rule is applied, the accuracy of DeepGroup increased dramatically compared to benchmarks after $\kappa \geq 3$.

Compared to the Sushi dataset, in all of the three Irish datasets, we observe a similar pattern and same level of accuracy value for both Borda and Plurality. In these datasets, we observe a higher prediction accuracy for smaller datasets. This means for Irish Dublin West, Irish Dublin North, and Meath we observe a higher accuracy for a fixed value of κ respectively. In all four datasets, and for both Borda and Plurality decision rules, our model indicates the same performance as Pop and RTCP and higher performance than O-Sim even for $\kappa = 1$.

These results suggest that as users participate more in various group decision-making processes, the model can more accurately learn their embeddings and consequently the embeddings of their groups. We observe that none of the benchmarks exhibit the same behavior since their performances remain almost steady as κ increases.

5.4.2 Decision Rules and Group Decision Prediction

To investigate the effect of different group decision rules on group decision prediction, we compare the accuracy of DeepGroup and baselines on group datasets generated



(a) Sushi

(b) Dublin West

(c) Dublin North

(d) Meath

Figure 5.3: The accuracy of DeepGroup and other benchmarks over different group decision rules for different group datasets generated on different preference datasets (a)–(d) with κ-participation method (fixed $\kappa = 5$).

with various decision rules. We use κ-participation group generation method with fixed $\kappa = 5$ and $n = 5000$. Figure 5.3 shows the accuracy of methods for various group decision rules (i.e., Borda, Plurality, and their mixtures) over different preference datasets.

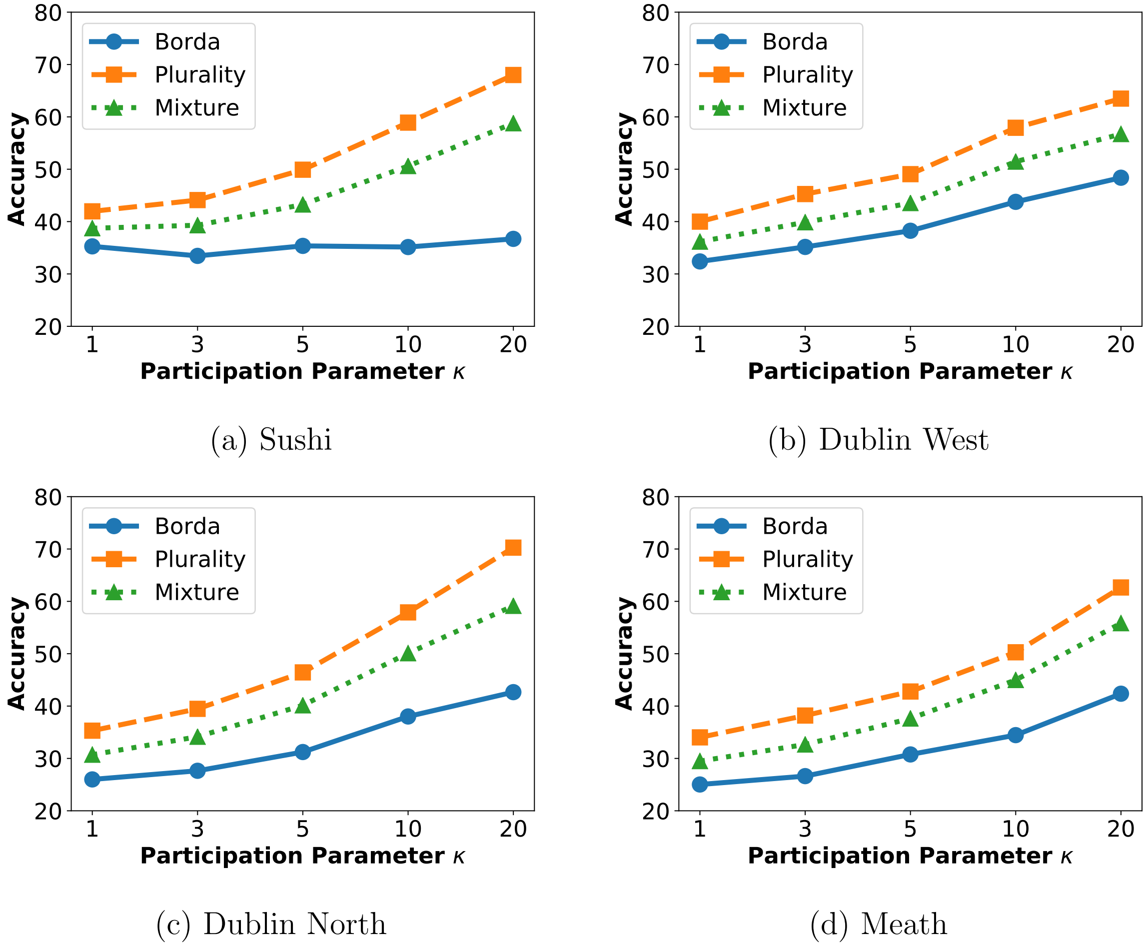
Investigating the overall accuracy of all the datasets, it is obvious that DeepGroup has the highest accuracy for Sushi when the group decision rule is Borda compared to other decision rules for all three group decision rules.

For all preference datasets, DeepGroup outperforms others over all decision rules to the various extent. It seems that DeepGroup offers the most improvement over baselines for plurality and the least improvement for Borda. One interesting observation is that DeepGroup still performs fairly well for the mixture of Borda and Plurality. This suggests that (a) DeepGroup does not necessarily require to be aware of group decision rules for successful prediction, and (b) DeepGroup can perform well when different groups use inconsistent decision rules.

5.4.3 Reverse Social Choice and Group Decision Rules

We study the accuracy of DeepGroup for reverse social choice (i.e., predicting individual preferences of group members) when various group decision rules are applied. We use κ-participation while varying κ over $\{1, 3, 5, 10, 20\}$. Figure 5.4 shows the accuracy of DeepGroup for various group decisions rules, preference datasets, and κ parameter.

For all group decision rules and preference datasets, the accuracy of the DeepGroup increases with participation factor κ. This implies that user personal prefer-



(a) Sushi

(b) Dublin West

(c) Dublin North

(d) Meath

Figure 5.4: The accuracy of DeepGroup for reverse social choice, group datasets generated by different group decision rules on various preference datasets (a)–(d) with κ-participation method.

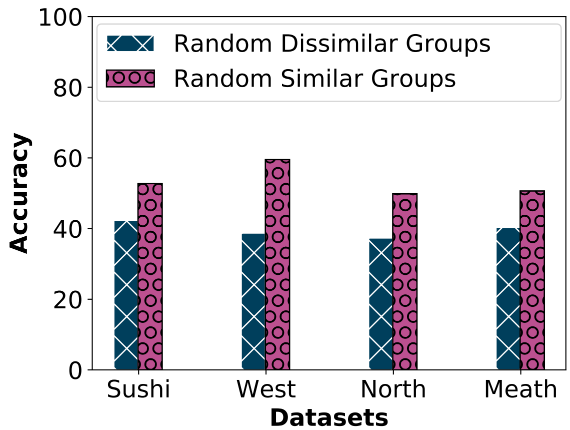
ences can be predicted more accurately if users participate in more group decisions.[4] Moreover, we observe that for the Irish Dublin North dataset the slope of accuracy line for plurality group decision rule is greater compared to the other datasets, and for $\kappa = 20$ DeepGroup can predict the individual preferences of group members up to 70% accurately.

One can observe an interesting pattern by comparing DeepGroup accuracy over group decision rules. For the Plurality decision rule, the accuracy is always the highest in all the datasets whereas Borda has the lowest accuracy. This observation is surprising: despite requiring the least preference data (i.e., only top choice) for decision making, plurality has the highest privacy leakage as the personal preferences can be predicted more accurately when it is deployed. In contrast, Borda has the lowest privacy leakage in this sense. Another important observation emerges from this experiment: when the decision rule is not inconsistent among the groups in a dataset (e.g., the mixture of plurality and Borda), DeepGroup is still effective in predicting the individual preferences.

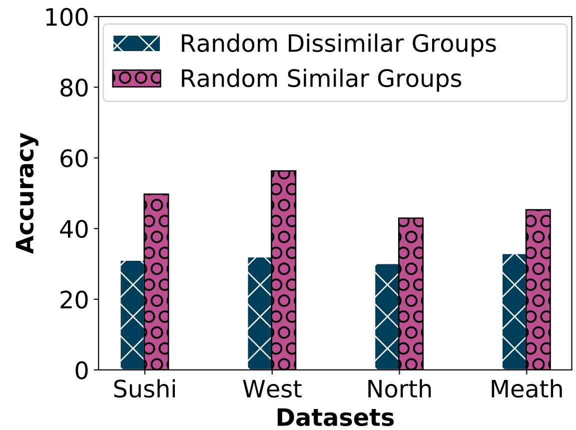
5.4.4 The Effect of Homophily and Heterophily

We aim to explore how the performance of DeepGroup changes when the group members possess similar or dissimilar preferences (homophily vs heterophily). To this end, we study the accuracy of DeepGroup for both group decision prediction and social reverse choice tasks, when the groups are generated by either of Random Similar Group (RSG) or Random Dissimilar Group (RDG). We fix the number of randomly generated groups $l = 1000$ for this experiment.

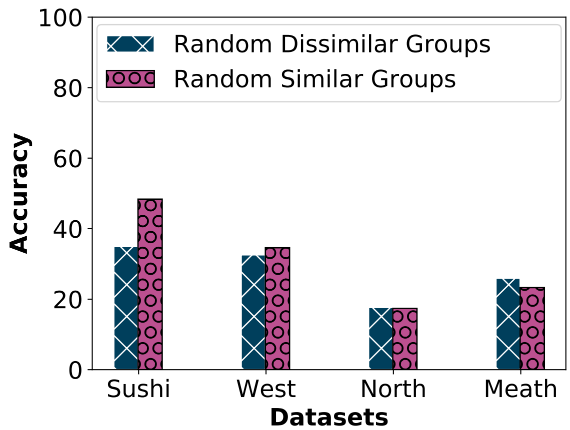
[4]We observed similar patterns for group decision prediction task.



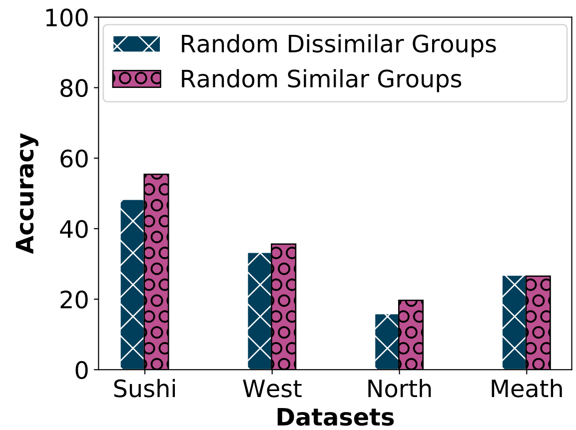
(a) Reverse Social Choice, Plurality



(b) Reverse Social Choice, Borda



(c) Group Decision, Plurality



(d) Group Decision, Borda

Figure 5.5: The accuracy of DeepGroup for similar and dissimilar random groups, group decision prediction and reverse social choice tasks, and Borda and Plurality group decision rules (a)–(d).

Figure 5.5 shows the accuracy of DeepGroup for various group generation methods, both prediction tasks, Borda and Plurality group decision rules, and different preference datasets. For the social reverse choice task, Irish Dublin West indicates higher accuracy compared to other preference datasets in both Borda and Plurality group decision rules. However, for the group decision prediction task Sushi reveals the highest accuracy for both group decision rules.

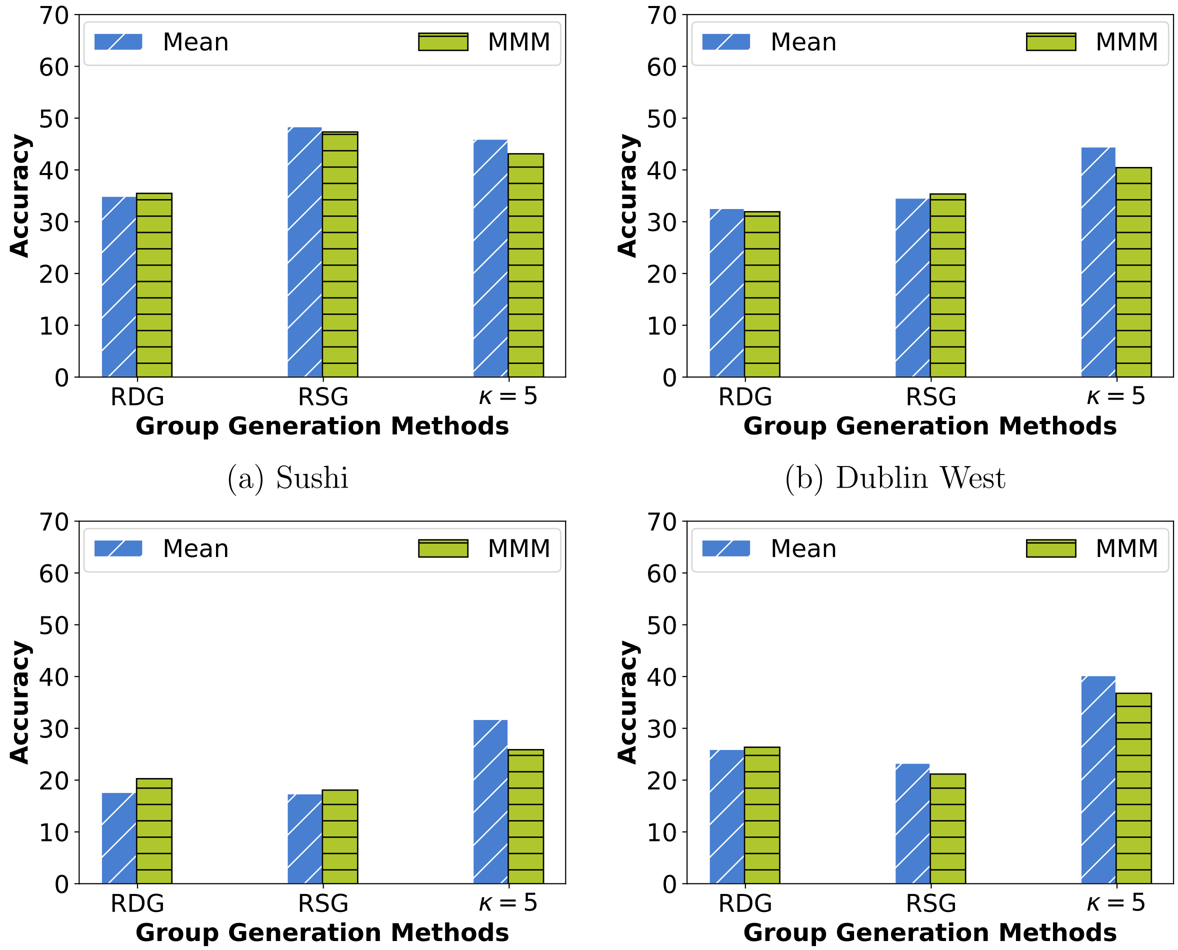
For both reverse social choice and group decision prediction, DeepGroup has higher performance for homophilic groups (i.e. group members have similar preferences) compared to heterophilic groups (i.e., groups with dissimilar preferences) regardless of the underlying group decision rule. From the privacy perspective, this result (especially for reverse social choice) implies that privacy-leakage of user preferences is the highest when the groups are composed of like-minded individuals. The rationale behind this observation is that a group's revealed decision is a good representative of all its group members' preferences when they are like-minded.

5.4.5 The Effect of Different Aggregator Functions

In this section, we aim to investigate the effect of different aggregator functions on group decision prediction of DeepGroup, mainly the Mean aggregator (Mean) and the combined Mean-Max-Min aggregator (MMM). To do so, we fix $\kappa = 5$, and the number of RDG, and RSG groups to 1000, and by considering both group decision rules we compare the performance of DeepGroup in different datasets.

In Figure 5.6 we compare the results of DeepGroup in different group generation methods when the group decision rule is plurality.[5] Although the Mean aggregator

[5]The results for Borda was qualitatively similar.



(a) Sushi

(b) Dublin West

(c) Dublin North

(d) Meath

Figure 5.6: The accuracy of DeepGroup for group prediction with Mean and MMM (aggregator functions), considering different group generation methods on various preference datasets (a)–(d), with Plurality group decision rule.

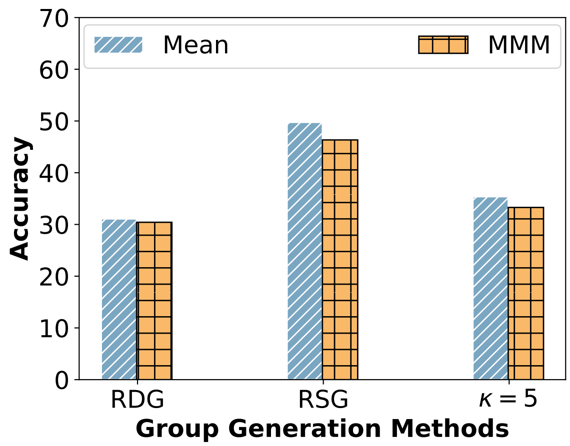
leads to higher accuracy for DeepGroup in most cases (specifically in κ-participation method), but MMM can indicate better performance in RDG group sets. This means in the case that the personal preferences of group members are not similar, MMM can learn better group representations. For RSG group sets, Mean and MMM perform closely. However, for Sushi and Irish Meath preference datasets, the accuracy of DeepGroup is higher with the Mean aggregator function. In contrast, the accuracy is higher when the aggregator function is MMM for Irish Dublin West and North.

To compare the prediction accuracy of DeepGroup with these two aggregator functions, for the reverse social choice task, we apply the same setting. We consider DeepGroup with both Mean and MMM aggregator as shown in Figure 5.7 for Borda decision rule.[6] We observe that the performance of DeepGroup with both Mean and MMM aggregator in the reverse social choice task are closely comparable. The exception is for the Sushi dataset where the performance of DeepGroup with Mean aggregator function is higher compared to that of MMM in RSG group sets.

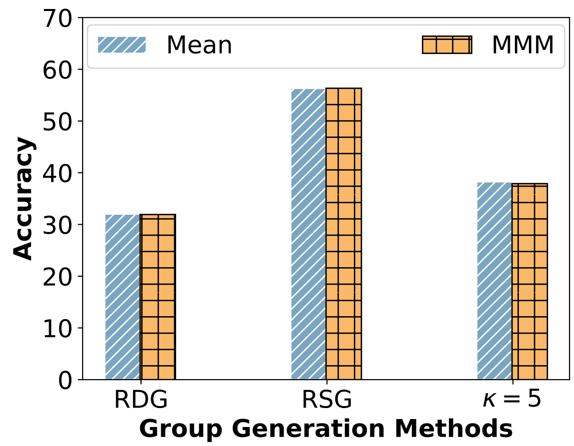
In another set of experiments, we aim to compare the prediction accuracy of DeepGroup for MMM and Mean aggregator functions in κ-participation group sets for both plurality and Borda group decision rules. For this purpose, we fix the number of users $n = 5000$ and vary κ over $\{1, 3, 5, 10, 20\}$, and we compare the accuracy of DeepGroup for both MMM and Mean aggregator functions in both group prediction and reverse social choice tasks.

As we observe in Figure 5.8, the overall performance of DeepGroup in group prediction task with both MMM and Mean are very comparable to each other. However, the performance of DeepGroup with the Mean aggregator is higher compared to the

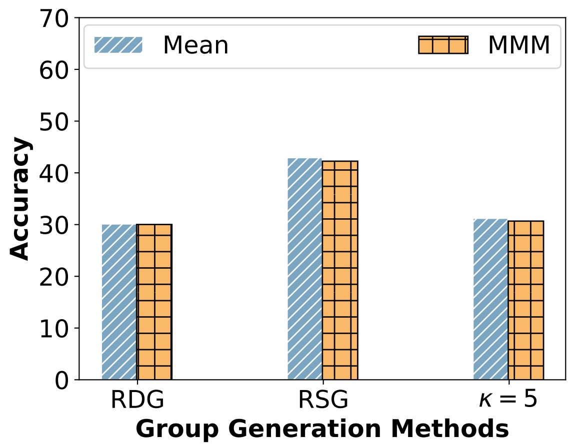
[6] The results for Plurality was qualitatively similar.



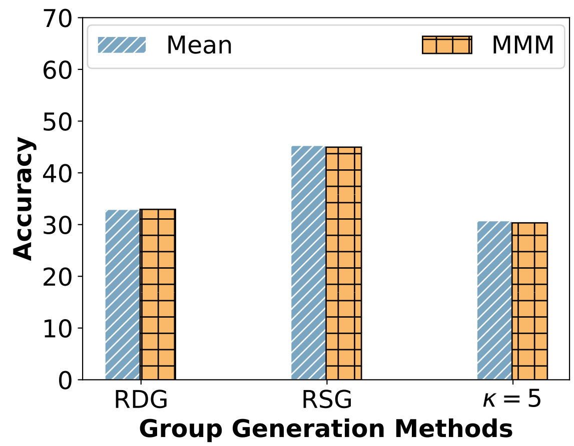
(a) Sushi



(b) Dublin West

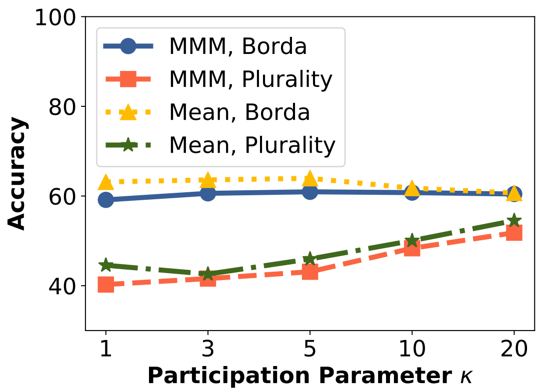


(c) Dublin North

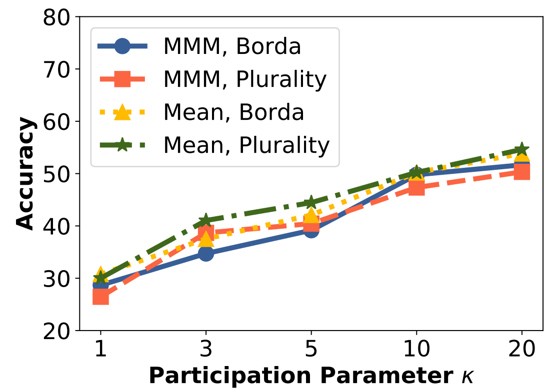


(d) Meath

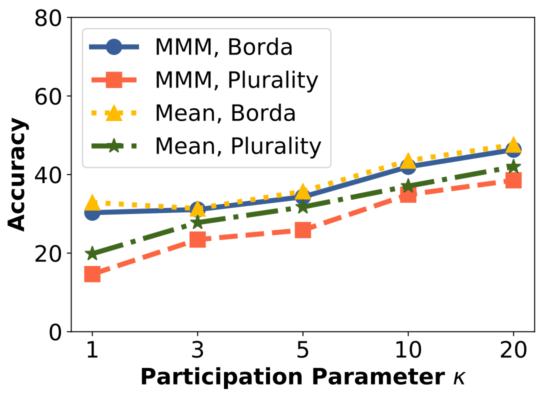
Figure 5.7: The accuracy of DeepGroup for reverse social choice with Mean and MMM (aggregator functions), considering different group generation methods on various preference datasets (a)–(d), with Borda group decision rule.



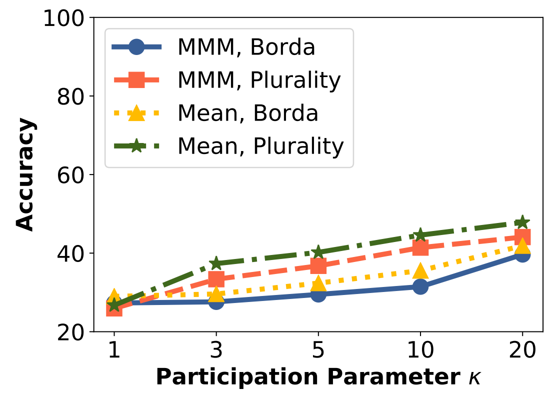
(a) Sushi



(b) Dublin West

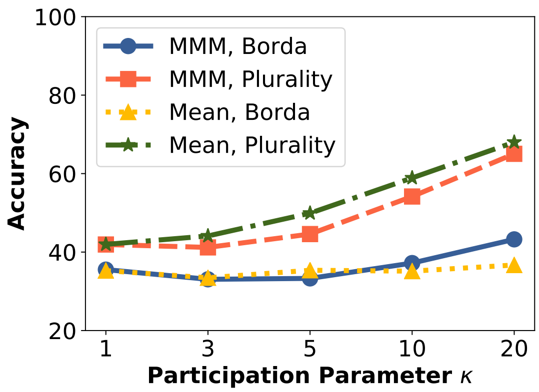


(c) Dublin North

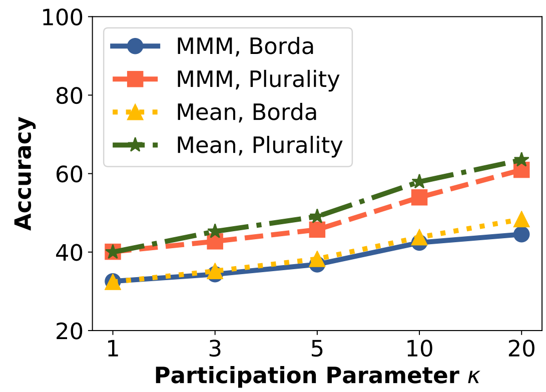


(d) Meath

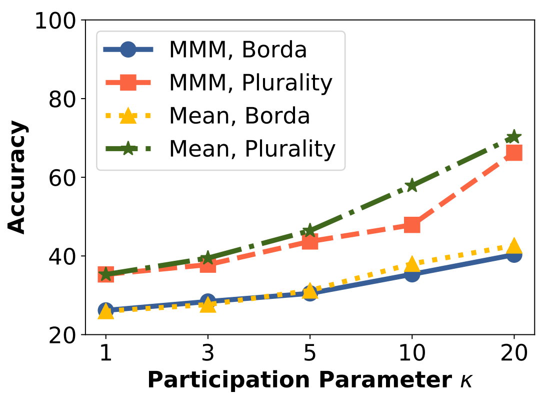
Figure 5.8: The accuracy of DeepGroup for group prediction task, with MMM and Mean aggregator functions on various preference datasets (a)–(d) with κ-participation method, and the plurality and Borda group decision rules.



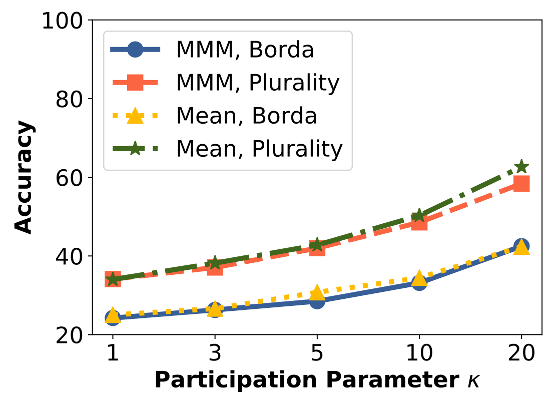
(a) Sushi



(b) Dublin West



(c) Dublin North



(d) Meath

Figure 5.9: The accuracy of DeepGroup for reverse social choice task, with MMM and Mean aggregator functions on various preference datasets (a)–(d) with κ-participation method, and the plurality and Borda group decision rules.

MMM aggregator function most times.

For the reverse social choice task, Figure 5.9 indicates that DeepGroup with MMM aggregator function performs better than the Mean aggregator in a low value of κ for both group decision rules. In most cases, by increasing the κ parameter, the Mean aggregator overtakes the MMM function.

Chapter 6

Conclusion

In this chapter, we review the conclusion of this thesis and future work. In section 6.1 we present the overall summary of our research and in section 6.2 we propose future directions of our work.

6.1 Overall Summary

In this work, we formulate the problem of group recommendation from group implicit feedback, with the goal of making item recommendation to a new group of users in the absence of personal user preferences. To address this problem, we introduce DeepGroup— a novel model for learning group representation based on observed group-item interactions. We conduct an extensive series of experiments to evaluate DeepGroup accuracy over various datasets and benchmarks while focusing on two special instances of our problem, reverse social choice and group decision prediction. Our findings confirm the effectiveness of DeepGroup in addressing these two problems. Our empirical results also show that different group decision rules (e.g., plurality, Borda, etc.) exhibit privacy leakage of concealed personal preferences with the various extent. Surprisingly, plurality, despite requiring less information than Borda, suffers more privacy leakage than Borda.

6.2 Future Directions

Although our proposed DeepGroup model is optimized by specific ranking datasets for the group and personalized recommendation tasks, it can be easily extended to different types of recommender systems. There are many fascinating directions to explore in future work. We highlight the most significant problems in the following sections:

6.2.1 Voting Rules

One can theoretically analyze some well-known voting rules in the context of our reverse social choice problem. There exist other scoring functions and aggregation strategies that may perform better compared to Borda or Plurality voting. Social relations between group members, fairness, and utility among group members are some significant examples that could be explored by applying other voting rules. These analyses can shed light on privacy-preserving characteristics of voting rules when the group decisions are publicly announced.

6.2.2 Improving Loss Function

Our DeepGroup model can possibly be improved by incorporating ranking loss functions and deploying more complex latent aggregator functions. There are well-known loss functions such as BPR [69] and Hinge loss functions that aim to optimize the model by learning the ranking among the input items. These pairwise personalized ranking loss function may improve the prediction accuracy by training positive and negative items together.

6.2.3 Utilizing Auxiliary information

DeepGroup is also a building block for the broader investigation of deep learning methods for group recommendation with group implicit feedback. Of practical importance is to extend the model with group and item features (e.g., descriptions, demographic information, etc.), side information (e.g., social networks between users), or context (e.g., time, location, etc.). By doing so, the model can learn more powerful representations for both groups and items which leads to higher prediction accuracy.

References

[1] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon, "Cats: A synchronous approach to collaborative group recommendation", in *Florida Artificial Intelligence Research Society Conference (FLAIRS)*, ser. FLAIRS'06, 2006, pp. 86–91.

[2] A. Crossen, J. Budzik, and K. J. Hammond, "Flytrap: Intelligent group music recommendation", in *Proceedings of the 7th international conference on Intelligent user interfaces*, ser. IUI'02, 2002, pp. 184–185.

[3] V. Rakesh, W.-C. Lee, and C. K. Reddy, "Probabilistic group recommendation model for crowdfunding domains", in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, ser. WSDM'16, 2016, pp. 257–266.

[4] S. Pizzutilo, B. De Carolis, G. Cozzolongo, and F. Ambruoso, "Group modeling in a public space: Methods, techniques, experiences", in *Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications*, ser. AIC'05, 2005, pp. 175–180.

[5] Z. Yu, X. Zhou, Y. Hao, and J. Gu, "Tv program recommendation for multiple viewers based on user profile merging", *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63–82, 2006.

[6] M. O'connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: A recommender system for groups of users", ser. ECSCW'01, 2001, pp. 199–218.

[7] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering", in *Proceedings of the 4th ACM Conference on Recommender Systems*, ser. RecSys'10, Barcelona, Spain, 2010, pp. 119–126.

[8] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu, "Group recommendation: Semantics and efficiency", *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 754–765, 2009.

[9] S. Seko, T. Yagi, M. Motegi, and S. Muto, "Group recommendation using feature space representing behavioral tendency and power balance among members", in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys'11, Chicago, Illinois, USA, 2011, pp. 101–108.

[10] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada, "Enhancing group recommendation by incorporating social relationship interactions", in *Proceedings of the 16th ACM International Conference on Supporting Group Work*, ser. GROUP'10, Sanibel Island, Florida, USA, 2010, pp. 97–106, isbn: 978-1-4503-0387-3.

[11] A. Salehi-Abari and C. Boutilier, "Preference-oriented social networks: Group recommendation and inference", in *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys-15)*, ser. RecSys'15, Vienna, Austria, 2015.

[12] L. Xiao, Z. Min, Z. Yongfeng, G. Zhaoquan, L. Yiqun, and M. Shaoping, "Fairness-aware group recommendation with pareto-efficiency", in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys'17, Como, Italy, 2017, pp. 107–115.

[13] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation", in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor MI USA, 2018, pp. 645–654.

[14] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman, "Practical voting rules with partial information", *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 1, pp. 151–182, 2011.

[15] L. Xia and V. Conitzer, "Determining possible and necessary winners under common voting rules given partial orders", Chicago, 2008, pp. 202–207.

[16] T. Lu and C. Boutilier, "Robust approximation and incremental elicitation in voting protocols", in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, ser. IJCAI'11, Barcelona, 2011, pp. 287–293.

[17] D. Hughes, K. Hwang, and L. Xia, "Computing optimal bayesian decisions for rank aggregation via mcmc sampling", in *UAI*, 2015, pp. 385–394.

[18] J. A. Doucette, K. Larson, and R. Cohen, "Conventional machine learning for social choice", in *Proceedings of the Twenty-Ninth AAAI Conference on Ar-*

tificial Intelligence, ser. AAAI'15, Austin, Texas: AAAI Press, 2015, pp. 858–864.

[19] J. Masthoff, "Group modeling: Selecting a sequence of television items to suit a group of viewers", *User Modeling and User-Adapted Interaction*, vol. 14, no. 1, pp. 37–85, 2004, ISSN: 0924-1868.

[20] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, Eds., *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[21] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič, *Group Recommender Systems: An introduction*. Springer, 2018.

[22] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation", in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, Macua SAR, China, 2019, pp. 566–577.

[23] Z. Huang, X. Xu, H. Zhu, and M. Zhou, "An efficient group recommendation model with multiattention-based neural networks", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, 2020.

[24] J. Bennett, S. Lanning, *et al.*, "The netflix prize", in *Proceedings of KDD cup and workshop*, New York, vol. 2007, San Jose, California, US, 2007, p. 35.

[25] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering", in *Proceedings of KDD cup and workshop*, vol. 2007, San Jose, California, US, 2007, pp. 5–8.

[26] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization", in *Advances in Neural Information Processing Systems (NIPS'07)*, Vancouver, B.C., Canada, 2008, pp. 1257–1264.

[27] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets", in *2008 Eighth IEEE International Conference on Data Mining*, Ieee, 2008, pp. 263–272.

[28] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback", in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa, Italy, 2016, pp. 549–558.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[30] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks", in *The 38th IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Vancouver, Canada, 2013, pp. 6645–6649.

[31] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks", in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, Bellevue, Washington, USA, 2011, pp. 129–136.

[32] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering", in *Proceedings of the 24th International Conference on Machine Learning*, ACM, Corvalis, Oregon, USA, 2007, pp. 791–798.

[33] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations", in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, ACM, Boston, MA, USA, 2016, pp. 17–22.

[34] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives", *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[35] J. Masthoff, "Group recommender systems: Combining individual models", in *Recommender Systems Handbook*, Springer, 2011, pp. 677–702.

[36] S. Dara, C. R. Chowdary, and C. Kumar, "A survey on group recommender systems", *Journal of Intelligent Information Systems*, vol. 54, pp. 271–295, 2020.

[37] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, and A. Aghasaryan, "Evaluation of group profiling strategies", in *Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-11)*, Citeseer, Barcelona, Spain, 2011.

[38] J. F. McCarthy and T. D. Anagnost, "Musicfx: An arbiter of group preferences for computer supported collaborative workouts", in *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW'98, 1998, pp. 363–372.

[39] J. F. McCarthy, T. J. Costa, and E. S. Liongosari, "Unicast, outcast & groupcast: Three steps toward ubiquitous, peripheral displays", in *International Conference on Ubiquitous Computing (UBICOMP 2001)*, Springer, Atlanta, Georgia, 2001, pp. 332–345.

[40] S. Berkovsky and J. Freyne, "Group-based recipe recommendations: Analysis of data aggregation strategies", in *Proceedings of the 4th ACM Conference on Recommender Systems*, ser. RecSys'10, Barcelona, Spain, 2010, pp. 111–118.

[41] P. Dwivedi and K. K. Bharadwaj, "E-learning recommender system for a group of learners based on the unified learner profile approach", *Expert Systems*, vol. 32, no. 2, pp. 264–276, 2015.

[42] A. Salehi-Abari, C. Boutilier, and K. Larson, "Empathetic decision making in social networks", *Artificial Intelligence*, vol. 275, pp. 174–203, 2019.

[43] A. Salehi-Abari and C. Boutilier, "Empathetic social choice on social networks.", in *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, Paris, France, 2014, pp. 693–700.

[44] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering", in *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 173–182.

[45] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations", in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, New York, USA, 2014, pp. 701–710.

[46] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks", in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, California, USA, 2016, pp. 855–864.

[47] A. Delic, J. Neidhardt, T. N. Nguyen, F. Ricci, L. Rook, H. Werthner, and M. Zanker, "Observing group decision making processes", in *Proceedings of the 10th ACM Conference on Recommender Systems*, Boston, Massachusetts, USA, 2016, pp. 147–150.

[48] T. De Pessemier, S. Dooms, and L. Martens, "Design and evaluation of a group recommender system", in *Proceedings of the 6th ACM Conference on Recommender Systems*, Dublin, Ireland, 2012, pp. 225–228.

[49] L. Xia, J. Lang, and M. Ying, "Strongly decomposable voting rules on multiattribute domains", in *AAAI*, vol. 7, 2007, pp. 776–781.

[50] K. Konczak and J. Lang, "Voting procedures with incomplete preferences", in *IJCAI-05 Workshop on Advances in Preference Handling*, Edinburgh, 2005, pp. 124–129.

[51] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans, "Constraint-based optimization and utility elicitation using the minimax decision criterion", *Artificial Intelligence*, vol. 170, no. 8-9, pp. 686–713, 2006.

[52] D. Braziunas and C. Boutilier, "Assessing regret-based preference elicitation with the utpref recommendation system", in *Proceedings of the 11th ACM Conference on Electronic Commerce*, Massachusetts, USA, 2010, pp. 219–228.

[53] N. Hyafil and C. Boutilier, "Mechanism design with partial revelation.", in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007, pp. 1333–1340.

[54] T. Lu and C. Boutilier, "Multi-winner social choice with incomplete preferences", in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI'13, Beijing, 2013.

[55] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives", *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[56] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems", in *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*, Florence, Italy, 2015, pp. 278–288.

[57] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide deep learning for recommender systems", in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, Boston, USA, 2016, pp. 7–10.

[58] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation", in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[59] M. Kula, "Metadata embeddings for user and item cold-start recommendations", *arXiv preprint arXiv:1507.08439*, 2015.

[60] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks", in *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, Cambridge, United Kingdom, 2017, pp. 495–503.

[61] Z. Zhu, J. Wang, and J. Caverlee, "Improving top-k recommendation via joint collaborative autoencoders", in *The World Wide Web Conference*, San Francisco, USA, 2019, pp. 3483–3482.

[62] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, 2016, pp. 353–362.

[63] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising autoencoders for top-n recommender systems", in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, San Francisco, USA, 2016, pp. 153–162.

[64] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems", in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, 2015, pp. 1235–1244.

[65] X. Li and J. She, "Collaborative variational autoencoder for recommender systems", in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, Canada, 2017, pp. 305–314.

[66] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering", in *Proceedings of the 27th International World Wide Web Conference*, Geneva, Switzerland, 2018, pp. 689–698.

[67] B. Askari, J. Szlichta, and A. Salehi-Abari, “Joint variational autoencoders for recommendation with implicit feedback”, *arXiv preprint arXiv:2008.07577*, 2020.

[68] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs”, in *Proceeding of the Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017, pp. 1024–1034.

[69] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback”, in *Conference on Uncertainty in Artificial Intelligence (UAI)*, Montreal, Canada, 2009, pp. 452–461.

[70] C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. D. Procaccia, and O. Sheffet, “Optimal social choice functions: A utilitarian view”, in *Proceedings of the 13th ACM Conference on Electronic Commerce (EC-12)*, Valencia, Spain, 2012, pp. 197–214.